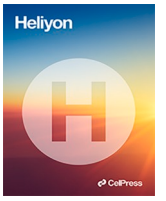
Heliyon 10 (2024) e31539

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

A CNN-LSTM model using elliptical constraints for temporally consistent sun position estimation

Mark Mpabulungi, Kyeongmin Yu, Hyunki Hong *

College of Software, Chung-Ang University, Heukseok-ro 84, Dongjak-ku, Seoul, 06973, Republic of Korea

ARTICLE INFO

Keywords:
Sun position estimation
CNN-LSTM
Deep learning
Elliptical constraints

ABSTRACT

More accurate sun position estimation could transform the design and operation of solar power systems, weather forecasting services, and outdoor augmented reality systems. Although several image-based approaches to sun position estimation have been proposed, their performance is significantly affected by momentary disruptions in cloud cover because they use only a single image as input. This study proposes a deep learning-based sun position estimation system that leverages spatial, temporal, and geometric features to accurately regress sun positions even when the sun is partially or entirely occluded. In the proposed approach, spatial features are extracted from an input image sequence by applying a separate Resnet-based convolution network to each frame. To ensure that the temporal changes in the brightness distribution across frames are considered, the spatial features are concatenated and passed on to a stack of LSTM layers prior to regressing the final sun position. The proposed network is also trained with elliptical (geometric) constraints to ensure that predicted sun positions are consistent with the natural elliptical path of the sun in the sky. The proposed approach's performance was evaluated on the Sirta and Laval datasets along with a custom dataset, and an R^2 Score of 0.98 was achieved, which is at least 0.1 higher than that of previous approaches. The proposed approach is capable of identifying the position of the sun even when occluded and was employed in a novel sky imaging system consisting of only a camera and fisheye lens in place of a complex array of sensors.

1. Introduction

Accurate sun position estimation is essential for improving the performance of solar power systems, weather forecasting services, and virtual object relighting in outdoor augmented reality (AR) systems. These applications can be widely classified based on whether fixed sun-tracking hardware was used [1,2] or not [3,4]. Setting up rigs for sun tracking applications involves cumbersome procedures, each of which may introduce errors and accuracies in the final estimated sun positions. Additionally, such systems are associated with prohibitive costs as well as extended initial setup and calibration times with every change in location and orientation. This significantly restricts their use in highly dynamic use cases.

Sun position estimation methods may be broadly classified into astronomical and image-based algorithms. First, astronomical computation-based approaches can be employed to compute the position of the sun for a specific location at a particular date and time. These require a highly accurate initial setup to specify rig location and orientation. They are also unable to distinguish between clear, cloudy, or overcast weather conditions.

* Corresponding author.
E-mail address: honghk@cau.ac.kr (H. Hong).

https://doi.org/10.1016/j.heliyon.2024.e31539
Received 26 September 2023; Received in revised form 10 May 2024; Accepted 17 May 2024
Available online 18 May 2024

Image-based methods are ideal for estimating the position and irradiance of the sun because sky images are a rich source of localized information regarding the distribution of brightness in the sky that occurs as a result of the sun's presence. The proposed approach also leverages these image features to reliably estimate the position of the sun even when it is partially or entirely occluded by cloud cover.

The radiance of the sun, as observed from a point on the earth's surface, varies with changes in time and location because of the earth's rotation, orbit, and axis of inclination [5]. Because images of the sky capture its current and transient state properties, they can be used to predict solar occlusion, weather conditions as well as changes to irradiance [6–8]. A wide range of approaches have been used to extract features from a single image and predict sun position or solar radiance [1,8–11]. However, atmospheric properties such as cloud cover and aerosol content cause ground-level solar irradiance to be highly variable [12]. The variability of these properties also affects the performance of sky image-based sun position estimation methods.

In Chu's approach [1], normalized color, saturation, and intensity values are merged into a single feature vector by performing simple numerical operations and then applying a threshold. The final sun positions are attained by computing the vertical and horizontal means of the values in the feature vector. Since it assumes that the sun can only exist in the brightest regions of an image, this approach is unable to reliably identify the position of the sun even when it is slightly occluded by clouds. In Hold-Geoffroy's approach [11], a standard feed-forward CNN with seven convolutional layers is employed in estimating the sun's position along with the sky and camera parameters of the Hošek-Wilkie illumination model [13]. This implies that it is capable of identifying the position of the sun even when it is partially occluded. However, since only a single image is used, its performance significantly deteriorates in cases where the sun is completely occluded. Its predictions over multiple frames may also be inconsistent with the natural path of the sun. For image inputs that contain high-intensity values, Paletta [9] performs sun position estimation by taking the vertical and horizontal median of the saturated pixels in an image. In order to obtain sun positions for cases where no high-intensity values exist, the trajectory of the sun is interpolated over a day and from previous days using polynomial regression. Although this approach attempts to address some of the limitations of Chu's approach [1], the interpolation operations it uses can only be performed with previously gathered data. This implies that it cannot be used for real-time sun position estimation tasks.

In order to achieve reliable sun position estimation suitable for high-output solar energy systems, spatial and temporal consistency needs to be considered. Therefore, this study proposes a sun position estimation system consisting of a real-time imaging system and a deep learning approach (with spatial and temporal considerations) for regressing sun positions from sky image sequences. In our real-time imaging system, sky images are captured with a fisheye lens and transferred to a computational unit that estimates sun positions (elevation and azimuth). Here, a deep learning approach is used to extract spatial features and leverage the temporal changes in the brightness distribution over image sequences to regress sun positions. In the proposed approach, multiple Convolution Neural Networks (CNNs) based on the Resnet architecture [14] are used to extract solar radiance-related features from a sequence of sky images. These features are concatenated and fed into a stack of Long Short-Term Memory (LSTM) layers that estimate the position of the sun even when it is partly or entirely occluded. The proposed CNN-LSTM network is trained using a loss term that computes the distance between predicted and ground truth points. Additionally, two extra constraint terms (elliptical consistency and elliptical shape penalties) are introduced to ensure that the proposed network's predictions are consistent with the sun's natural path,.

The proposed method was trained and evaluated using images from the Laval [15] and Sirta [2] sky image databases, as well as a custom dataset (CAU-2). Evaluation results show that the proposed method predicts sun positions more accurately than previous methods. The main contributions of this manuscript are highlighted as follows.

- A real-time imaging system for sun position estimation that can easily be installed because it only requires Global Positioning System (GPS) information during its initial setup. Since no additional equipment, such as aero lasers or infrared cameras, is necessary, the proposed imaging system is suitable for dynamic use cases.
- A CNN-LSTM model that extracts spatial features and leverages temporal changes in the brightness distribution across sky image sequences to regress sun positions. This implies that the proposed method can reliably estimate sun positions even with significant changes in the level of cloud cover over consecutive frames by leveraging features from current and previous frames.
- Elliptical consistency and shape constraints that ensure that predicted sun positions are more consistent with the natural path of the sun over a day. This implies that even when no discernible brightness distribution is present, predicted sun positions are plausible (not irregular).

2. Theory

2.1. Sun position estimation using astronomical methods

The position of the sun in the sky, as seen from a given point on the Earth's surface at a specific date and time, can be estimated by employing various astronomical algorithms. Walraven [16] simplified the high-precision calculations used to generate the American ephemeris and nautical almanac solar tables by reckoning time from a closer date. In this approach, the earth was considered to be at the center of a celestial sphere, with celestial objects like the sun orbiting it. Michalsky's Almanac algorithm employed a more straightforward form of Walvaren's approach with adjusted angular ranges for the coordinate system [17]. It achieved higher accuracy by also considering variations in the length of a solar day, refraction in the atmosphere, as well as the angular extent of the sun at sunrise and sunset in addition. A key limitation of these approaches is that they are inaccurate, with their validity ranging from 15 to 100 years.

Meeus used the French planetary theory (Variations Sèculaires des Orbites Planètaires Theory: VSOP87) to accurately model

planetary motion, extending the computed sun position's validity to 8000 years (2000 BC to 6000 AD) by considering planetary perturbations and gravitational interactions [18]. Grena's sun pose algorithm (SPA) [19] was a refinement of the complicated steps described in Meeus' SPA [18], with a focus on the sun and not the planets or stars. A key limitation of these approaches is that they rely heavily on the orientation of sun position estimation rigs and the accuracy of GPS sensors. This implies that even minor errors in the orientation or location of the rig translate to significant errors in the computed sun position. Additionally, they are not suitable for relighting or irradiance applications because they are unable to distinguish between clear, cloudy, or overcast weather conditions.

2.2. Image-based sun position estimation

Sky images are a rich source of localized information regarding the overall state (brightness distribution as a result of the sun's presence) and transient characteristics of a visible sky. In Lalonde's approach [20], an input image is segmented into ground, vertical surface, and sky pixels based on visual cues such as color, texture, and perspective distortions. A normal probability distribution indicating sky luminance is generated from the segmented sky pixels by regressing the parameters of the Perez sky model [21] for clear, partially cloudy as well as entirely overcast scenes. Because the sun is the primary light source in outdoor scenes, the position and orientation of shadows on the ground plane were used to determine the position of the sun. A major drawback of this approach is that it requires accurate shadow lines and well-segmented scene elements (ground, vertical surface, and sky regions).

In Chu's approach [1], a binary feature map was generated from the RGB and HSV color spaces of an input image. Here, the average of an input image's hue and saturation channels was subtracted from that of its primary (RGB), intensity, and value channels. Then, a thresholding operation was performed, resulting in a binary feature map. In order to identify the position of the sun, the vertical and horizontal means of the generated feature map were computed. Multilayer Perceptrons (MLPs) were used to forecast direct normal irradiance values. A significant shortcoming of these approaches is that they are unable to handle complex real-world scenes with cloudy and overcast weather conditions.

In order to facilitate the realistic placement of virtual objects in real-world outdoor scenes, Hold-Geoffroy proposed a CNN-based technique to estimate the parameters of an outdoor illumination model from a single panorama image [12]. Here, seven CNN layers followed by a fully connected layer feed two separate heads: one for estimating the position of the sun and the other for estimating the sky and camera parameters of the Hošek-Wilkie illumination model [13], which was modified to account for sky radiance. The sun position head outputs a probability distribution with a likelihood value indicating the presence of the sun in each of the 160 bins that were used to represent the entire sky hemisphere. A reconstruction error is computed using ground truth and predicted turbidity, exposure, and sun position values. However, the turbidity parameter of the sky model used restricts this approach's representational accuracy to only clear skies. This implies that its accuracy degrades as cloud cover increases.

In Rahim's approach [22], the Hough gradient method is used to identify the circles (sun-like regions) in an image. Here, circle candidates are generated by voting in the Hough parameter space and then selecting local maxima in an accumulator matrix. A major limitation of this approach is that several false circles or even no circles may be detected as a result of partial or entire occlusion of the sun by cloud cover. In Paletta's approach [9], the visible sun regions in an input image are localized by considering the brightness of pixels relative to surrounding pixels. In cases where the sun is not visible, the sun's trajectory is determined by interpolating over past observations taken at the same time on previous days and through observations made from the start of the current day. The maximum pixel value in an image is used to determine whether the sun was visible in the image. A fundamental limitation of this approach is that it requires observations to be continually stored, which significantly increases the setup time and memory required. Additionally, observations missing due to an extended period of cloudy days may affect the quality of estimation results.

2.3. Solar irradiance estimation and forecasting methods

The solar irradiance incident on a surface largely depends on cloud cover and estimating it from sky images is currently an active area in solar energy-related studies. Zhao used a 3D CNN to extract textural and temporal cloud features from a sequence of cloud images [10]. These features were then fed into a linear autoregressive model and a nonlinear MLP, capable of generating accurate solar radiance forecasts up to 10 min ahead. In Feng's study on solar irradiance forecasts, several frames from a sky image sequence were stacked vertically and horizontally into a single 2D image [7]. Visual Geometry Group (VGG) based models were then used to regress highly accurate solar irradiance forecasts from the 2D image. In the proposed CNN, features were extracted using five feature learning blocks. Each feature learning block had two or three convolution layers and a max pooling layer. The extracted features were then passed to a stack of fully connected layers that generated solar output predictions. Siddiqui achieved solar irradiance forecasts of up to 4 h ahead by employing dilated convolutions, as well as auxiliary data such as air temperature, wind speed, relative humidity, and barometric pressure [8]. Here, features were extracted from multiple sky video frames using dilated convolutions. A two-tier LSTM network was then used to generate solar output predictions from the extracted features.

2.4. Relighting virtual objects in outdoor AR scenes

Realistic lighting is essential for maintaining visual consistency when virtual objects are integrated into real-world scenes. The sun is the primary light source in outdoor scenes, so detecting its position is necessary for relighting virtual objects. In order to realistically light virtual objects, Chen used intrinsic properties extracted from a single image to estimate the parameters of a ray-based illumination model [23]. Here, regression-based coarse scene understanding models were used to decompose an input image into its shading, geometry, reflectance, and semantic components. After applying RANdom-Sample Consensus (RANSAC) refinement to an aggregation

of these components, the Levenberg Marquardt algorithm was used to estimate the intensity of a fixed number of light sources modeled in the sparse radiance map. Hold-Geoffroy proposed a method to recover plausible illumination from a single outdoor image with a limited field of view g. In this study, an auto-encoder-based architecture was trained on datasets consisting of sky and scene images. Then, the auto-encoder was used to convert a Low Dynamic Range (LDR) panorama dataset to a High Dynamic Range (HDR) one. Two image encoders were trained to map ground scenes to their corresponding skies with sun positions. These mappings were then used to generate the lighting information for visually realistic virtual object synthesis.

3. Materials and methods

This study introduces a deep learning-based sun position estimation system that uses sky image sequences. In the proposed approach spatial, temporal, and geometric features are considered in regressing sun positions from image sequences. Multiple convolution networks extract spatial features from image sequences, and a stack of LSTM layers identifies the temporal changes in the brightness distribution across frames to regress sun positions. Since the sun takes an elliptical path in the sky, the proposed network is trained with elliptical (geometric) constraints that ensure that predicted sun positions are consistent with the sun's natural path. In order to facilitate the use of the proposed approach in dynamic use cases, a novel sky imaging system, consisting of only a camera and fisheye lens, is also presented. Fig. 1 shows our real-time imaging system and the proposed network. Estimated sun positions are transmitted to client devices (solar power systems, sun trackers, and mobile devices) using the Hypertext Transfer Protocol (HTTP).

3.1. Sky imaging system

In Chu's approach [1], a multi-filter rotating shadow band radiometer, two fisheye cameras, and an automatic solar tracker were used to identify the sun's position and monitor solar radiance. However, this setup is impractical for dynamic scenarios with frequent movement because it is relatively expensive and cumbersome. In contrast, our imaging system captures sky images with a Canon EOS C70 camera and a Canon EF zoom lens, both of which are commercially available. Here, the neutral density filter was set to 8, and a focal length of 8 mm was used. More specifically, the Canon EOS C70's wide 4K 35 mm DGO imaging sensor and its ND filters allow it to deliver reasonable sky image quality under a wide range of illumination conditions. This makes it ideal for dynamic use cases. The Canon EF zoom lens provides a 180° field of view, making it capable of capturing fish-eye images of the entire sky.

The real-time imaging system passes frames to a control unit using a Vention 8K fiber optic High-Definition Multimedia Interface (HDMI) cable connected to a Universal Serial Bus (USB) HDMI capture cable. The AP-HDC4K capture cable receives a YUV format 4K video signal through an HDMI cable via the transition minimized differential signaling (TDMS) protocol and retransmits it as packets via USB to the control unit at 60 frames per second.

Our sky imaging system has a simple setup consisting of a single commercially available omnidirectional camera with a fish-eye lens and a panoramic tripod head. During the initial setup, GPS information is used to align the primary axes of the proposed imaging system's coordinate system with those of Earth's spherical coordinate system. This makes the proposed approach ideal for dynamic outdoor applications where rig position and orientation may be frequently altered, such as light source estimation for solar power and augmented reality systems, as well as localized weather forecasting and building orientation optimization. As shown in Fig. 1, this makes computing and transmitting real-time light-rendering information to client devices feasible.

3.2. Sun position estimation using CNN-LSTM

The barrel distortion effects associated with images captured using fisheye lenses significantly affect the performance of CNNs as well as other 2D image-based feature extraction approaches [24,25]. In order to address this limitation, Bourke's approach [26] was used to convert the fisheye images provided in the datasets into their equirectangular form prior to training and inference. More

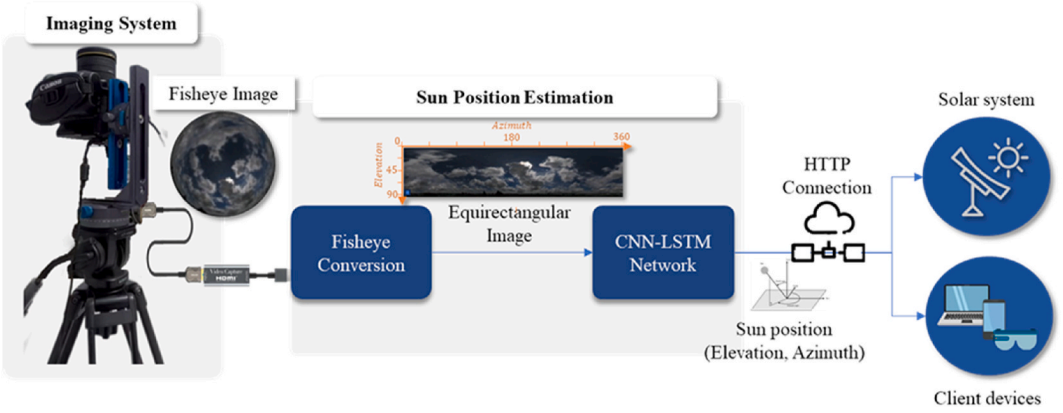


Fig. 1. Sun position estimation system architecture with an HTTP-based data transmission setup.

specifically, a precomputed look-up table, which maps each pixel in the fisheye image to a corresponding pixel in the equirectangular image space, was generated. This facilitated real-time (about 10 ms) fisheye to equirectangular conversions.

In the proposed CNN-LSTM architecture, the ResNet50V2 [14] network was extended and used to extract solar radiance-related features from each frame of the input image sequence, as shown in Fig. 2. This was because its vast generalization power, as demonstrated in several studies [14,27], could be excellent for extracting the local spatial relationships between the clouds and the radiance distribution created by the sun. It consists of a single convolutional layer (Conv) and four convolution blocks (Conv Block). The number of convolutional layers in the original architecture and their corresponding convolution filter sizes, strides, and paddings are preserved, but batch normalization, max pooling, and fully connected layers are appended to the end of the network. This architecture was employed for 96 × 386 and 128 × 512 sized image inputs in our experiments. A 3 × 13 × 2048 sized output is flattened into a 79,872-length vector for 96 × 386 image inputs, while a 4 × 16 × 2048 sized output is flattened into a 131,072-length vector for 128 × 512 image inputs. Then, an FCN layer generates the final vector embedding of length k from each frame in the input sequence. In our experiments, setting k to 64 for smaller images and 128 for larger images was found to be sufficient for generating robust image features.

In order to model the temporal relationships between the frames, the generated encodings are concatenated to form a shape of 64 × *N* and fed into a stack of 3 LSTM layers. N represents the number of input frames. LSTM models have been used in several sequence modeling tasks, such as inter-language translation [28,29], voice activity modeling [30,31], and identifying disaster-related posts on social media [32], because they are capable of effectively capturing long temporal dependencies without suffering from vanishing gradients. Hochreiter's LSTM [33] employs an efficient gradient-based algorithm to train an architecture (Fig. 3) that enforces constant error flow. In Eq. (1) to Eq. (3), W_i, W_c, and W_f represent the weight matrices used in computing the input cell's output (it), the new candidate values (Ct), and the forget gate's output (f_t), respectively. b_i, b_c, and b_f represent the bias values added to the products of the weight matrices (W_i, W_c and W_f) and the cell's previous hidden state (h_{t-1}) as well as the current input value, x_t in the weight update function ($W_i \cdot [h_{t-1}, x_i] + b_c$) for Eq. (1). In order to decide the values to be updated in the input gate (Eq. (1)) and forget gate (Eq. (3)), the output of the weight update function is fed into a sigmoid function, σ (), in each case. In Eq. (2), a vector of new candidate values is generated using a hypertangent layer, tanh(). In order to obtain the new cell state (C_t), the previous cell state (C_{t-1}) is multiplied by the output of the forget gate and added to the product of the input gate's output and the new candidate values, as shown in Eq. (4). This is passed to the output gate, which protects other units from perturbation by ensuring that only relevant memory contents are stored in the central unit. Since vanilla LSTMs have been shown to perform reasonably well on various tasks [34], the extracted features are concatenated and passed to three vanilla LSTM layer configurations that estimate sun positions, even when partly or entirely occluded.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i). \tag{1}$$

$$\widetilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{2}$$

$$f_t = \sigma\left(W_f \bullet [h_{t-1}, x_t] + b_f\right) \tag{3}$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t . \tag{4}$$

Since the Earth rotates on its axis and takes on an elliptical path around the sun, the sun follows a consistent path in the sky, as observed from a point on the Earth's surface. LSTM layers are well suited for identifying and leveraging the temporal changes in the

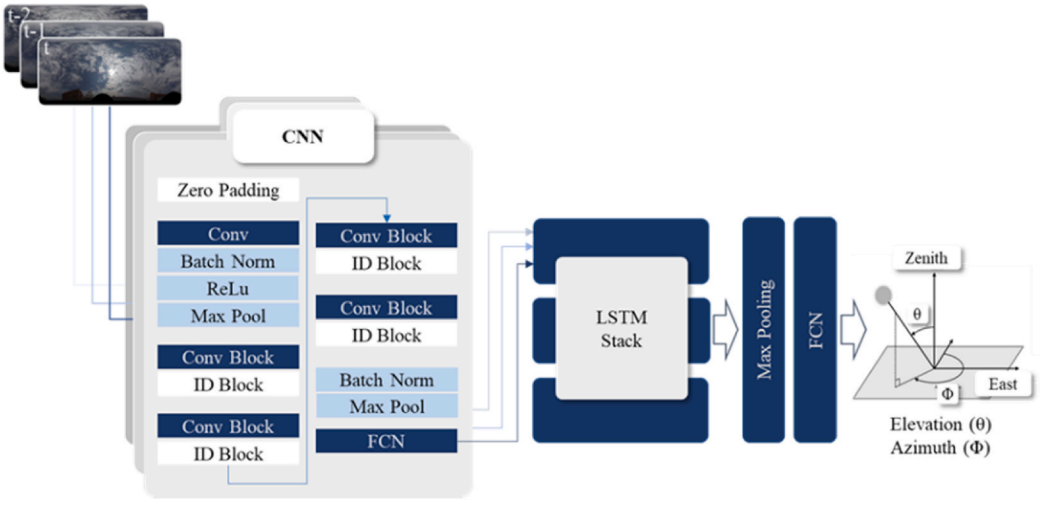


Fig. 2. Architecture of the sun position estimation network with each CNN model encoding a single image and a stack of LSTM layers, max pooling, and fully connected layers predicting sun positions.

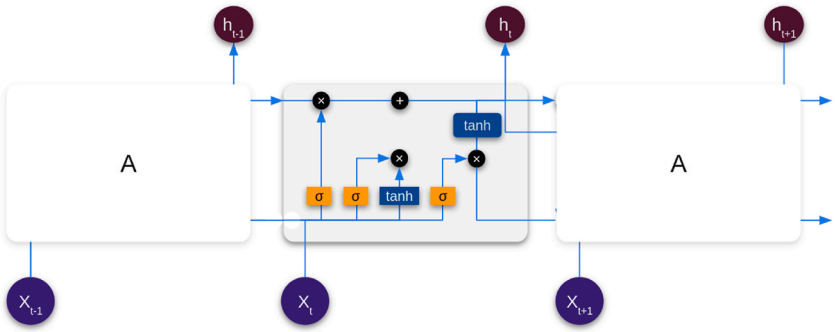


Fig. 3. Architecture of single cell used in vanilla LSTM networks [33].

brightness distribution over consecutive frames.

After performing a max pooling operation, an FCN layer at the very end of the network (Fig. 2) regresses the final sun positions (elevation and azimuth) of shape, 2 × N. Here, N represents the number of input frames.

3.3. Losses and elliptical constraints

While training the proposed CNN-LSTM network, the mean absolute error (e_m) between predicted and ground truth sun positions is computed as an $L1$ norm. In order to amplify the impact of small and medium-sized errors, Feng's piece-wise Wing loss [35] was used as the primary loss function. As shown in Eq. (5), a log function is applied when the mean absolute error (MAE) computed is smaller than the threshold, r [35]. Otherwise, the MAE is used in its plain form. The impact of various error ranges on gradient descent can be described by a V-like shape, with higher error values having larger impacts than lower ones. Eq. (5) increases the curvature of the V-like shape to emphasize small and medium error ranges.

$$L_w = \begin{cases} rln\left(1 + \frac{|e_m|}{\epsilon}\right) & if\ |e_m| < r \\ |e_m| - c, otherwise \end{cases}, \tag{5}$$

where r determines the range of loss values covered by the non-linear part of the V-like shape, ϵ controls its curvature and $c = r - rln\left(1 + \frac{w}{\epsilon}\right)$ is a constant that smoothens the transition between linear and non-linear parts.

As shown in Fig. 4, although the path of the sun in the sky over a given day may vary in length and orientation, it always takes on an elliptical form. This path can, therefore, be expressed using the quadratic equation for $F(x, y)$ in Eq. (6), which describes the position and orientation of an ellipse in 2D space [36]. Here, x and y represent an ellipse's horizontal and vertical components. The ellipse coefficients are $a, b, c, d, e,$ and f. In order to ensure that the proposed network's predictions are consistent with the natural path of the sun, two extra constraint terms (elliptical consistency and elliptical shape penalties) that penalize non-elliptical sun positions have been added to the primary loss computation to form the final loss in our training procedure.

First, the elliptical consistency penalty (C_c) is computed by comparing the geometric properties (center, width, height, and orientation) of the ellipses of best fit generated from ground truth and predicted sun positions. Larger C_c values imply that the ellipse generated with predicted points dramatically differs from that generated with ground truth points. We were thus able to examine the

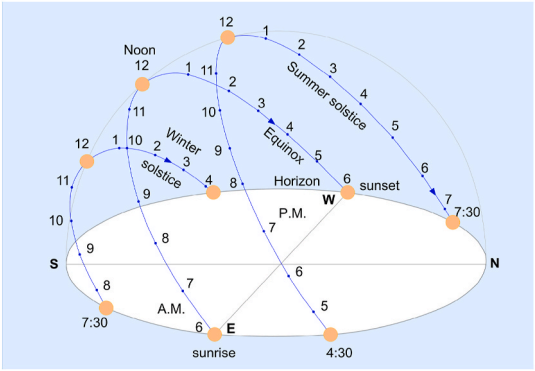


Fig. 4. Isometric drawing showing the elliptical path of the sun in the sky on various days in the year.

differences between the elliptic geometries of predicted and ground truth points.

$$F(x,y) = -ax^2 + bxy + cy^2 + dc + ey + f. \tag{6}$$

Because Flusser's method [36] is numerically stable and guarantees ellipse-specific solutions even with noisy data, it was used to estimate the ellipse coefficients in Eq. (6) for a given sequence of predicted values. To simplify the process of finding an ellipse's center, width, and height from its equation coefficients, the numerator (V_n) and denominator (V_{d_1}, V_{d_2}) values are precomputed with Eq. (7), Eq. (8), and Eq. (9) [36].

$$V_n = af^2 + cd^2 + eb^2 - 2bdf - ace. \tag{7}$$

$$V_{d_1} = \left(b^2 - ac\right) \times \left(\sqrt{4b^2 + a^2 - 2ac - c^2} - (a + c)\right). \tag{8}$$

$$V_{d_2} = \left(b^2 - ac\right) \times \left(-\sqrt{4b^2 + a^2 - 2ac - c^2} - (a + c)\right). \tag{9}$$

The height (h), width (w), and center (x_0,y_0) of the ellipse can then be computed using Eq. (10) and Eq. (11). The orientation (θ) of the ellipse is calculated using Eq. (12) [36].

$$h = \sqrt{\frac{2V_n}{V_{d_2}}}, w = \sqrt{\frac{2V_n}{V_{d_1}}}. \tag{10}$$

$$x_0 = \frac{cd - bf}{b^2 - ac}, y_0 = \frac{af - bd}{b^2 - ac}. \tag{11}$$

$$\theta = \frac{1}{2} tan^{-1}\left(\frac{2|b|}{|a| - |c|}\right). \tag{12}$$

The center, width, height, and orientation of an ellipse are placed in an ellipse property vector, $\overrightarrow{\boldsymbol{E}} = \left[x_0, y_0, w, h, \theta\right]^T$. The elliptical consistency constraint (C_c) is then calculated using Eq. (13). The squared distance between the centers, heights, and widths of the predicted and ground truth ellipses is computed. The square distance and angular differences are normalized using the maximum distance and maximum value in the angular distance space, respectively, and then combined as shown in Eq. (13).

$$C_c = \sum_{i=1}^{5} \left(E_{GT_i} - E_{Pred_i}\right)^2. \tag{13}$$

Second, the elliptical shape penalty (C_s) is a measure of how close predicted points are to their ellipse of best fit. In other words, it is intended to ensure that predicted points are as close as possible to their ellipse of best fit. If a C_s value of zero is achieved, then all predicted values lie along the edge of their ellipse of best fit. Determining how close predicted points are to an ellipse of arbitrary position and orientation is very computationally intensive. In order to address this, the ellipse that best fits predicted points is shifted to the origin, and the rotation matrix ($\boldsymbol{R}$) is used to align its major and minor axes as well as its corresponding points with the vertical and horizontal axis [37]. Here θ represents the orientation of the ellipse of best fit, which is computed using Eq. (14).

$$R = \begin{pmatrix} cos\,\theta & -sin\,\theta \\ sin\,\theta & cos\,\theta \end{pmatrix}. \tag{14}$$

Eq. (15) computes the distance between an arbitrary point, $E(E_x,E_y)$ on the axis aligned ellipse of best fit and a predicted point, $P(P_x, P_y)$ rotated about the same angle [38]. Eq. (16) is computed iteratively and minimized using the Newton-Raphson optimization method [38] until a value with acceptable tolerance is found. Here , m_{n+1} is the distance between a predicted point and its corresponding ellipse of best fit at the $(n+1)^{th}$ step, starting with an initial guess (m_0). In this optimization process, the accuracy of the final estimate is largely dependent on the initial guess (m_0). In order to ensure that the optimization converges on a global solution even when points are inside the ellipse, the initial value (m_0) is set using polygon vertices generated around the ellipse.

$$D = \begin{pmatrix} P_x\,cos\,\theta \\ P_y\,sin\,\theta \end{pmatrix} - \begin{pmatrix} E_x\,cos\,\theta - E_y\,sin\,\theta \\ E_y\,sin\,\theta + E_y\,cos\,\theta \end{pmatrix}. \tag{15}$$

$$m_{n+1} = m_n - \left(\frac{dD^2}{dm} \div \frac{d^2D^2}{dm^2}\right). \tag{16}$$

The final loss (L) of the proposed CNN-LSTM model is computed by summing the Wing loss (L_w), which is an extension of the mean absolute error, the elliptical consistency penalty (C_c), and the elliptical shape penalty (C_s) as shown in Eq. (17). Here, λ_1, λ_2, and λ_3 are set to 0.65, 0.175, and 0.175, respectively, as guided by previous experiments.

$$L = \lambda_1 L_w + \lambda_2 C_c + \lambda_3 C_s. \tag{17}$$

4. Results and analysis

4.1. Dataset

The proposed sun position estimation network was trained and tested using the Laval [15] and Sirta [2] datasets as well as a custom dataset (CAU-2). Table 1 details the number of images, resolutions, and ground truth labels available in each dataset. The year when images were captured is also indicated for each dataset. In the Laval and Sirta datasets, the sky was captured under a wide range of weather conditions, including clear, partially cloudy, and overcast skies with varying levels of rainfall. The CAU-2 dataset contains images of overcast skies but no scenes of heavy rainfall.

The Sirta dataset consists of 253,431 fisheye sky images captured in 2017 at the Sirta observatory in Paris, France. The images were captured using a CMS Ing. Dr. Schreder Gmbh cloud cam II with an exposure time of January 2000 s, an aperture value of f/8.0, a shutter speed of January 2000 s, and a focal length of 5.8 mm. Images of two exposures are provided, but only the lower-exposure images were used in our experiments to avoid severe over-exposure effects. Because no sun position labels are provided with the Sirta dataset, ground truth sun positions were generated from the camera location and image metadata using Meeus' SPA algorithm [18]. Aside from the focal length, none of the intrinsic fisheye camera parameters that describe the nature and extent of the radial distortion were provided in this dataset. Equi-rectangular images were generated by adapting the spherical projection model to the geometry of the fisheye camera, which results in slight camera distortions in regions near the edge of fisheye images. Since the sun usually appears at the edge of the fisheye image captured at sunset and sunrise, only images with a time stamp between 9 a.m. and 3 p.m. were used in our training and testing. This also addresses the occlusion of the sun by surrounding buildings, structures, and trees in the early morning as well as late evening. Colorfulness [39] and perceived luminance [40] measures were used to identify unusable images. Days with a high proportion of unusable images were excluded.

The Laval dataset consists of HDR sky dome panoramic maps with a resolution of 1024 × 2048 and corresponding sun positions (zenith and azimuth angles). Since panoramic images were provided, there was no need to consider the camera model or its intrinsic properties. The CAU-2 dataset comprises 1777 fisheye images captured at Chung-Ang University in Seoul, South Korea, using the imaging system described in subsection 2.1. A 6.0 mm focal length and f3.2 aperture were used. The ISO was set to 200, and the ND Filter to 8. Shutter speeds of 1/30 and 1/60 were used. Ground truth labels were generated using an approach similar to that used for the Sirta dataset. In order to obtain the fisheye camera parameters, Kannala's method [41] was used for calibration. Each dataset was split into training, validation, and test batches using a split of 65 %, 25 %, and 10 %, respectively. Due to the limited memory and processing power available, images were resized using bilinear interpolation.

4.2. Performance of the sun position estimation networks

In order to evaluate the performance of the proposed method under various input and model configurations, experiments were conducted on a computer running TensorFlow 2.8 and equipped with an Intel® Core™ i9-10920X CPU and an Nvidia RTX 3090 graphics processing unit.

First, the BGR format input image, received by the control unit, is converted to RGB format. In cases where a gradient map is used, pixel values are averaged across the three channels to form a grayscale image. A Gaussian blur operation with its standard deviation set to 1.5 is then performed on the grayscale image prior to vertical and horizontal gradient computations. The frames are stacked up to a particular frame length, with the last frame at the top. In order to normalize pixel values and ensure their intensity is between 0 and 1, all pixel values are then divided by 255.

As shown in Table 2, the impact of using gradient maps as additional inputs to both CNN and CNN-LSTM models was examined using images from the CAU-2 dataset. For all CNN-LSTM-related experiments conducted in this study, each LSTM layer is initialized with a Glorot normal distribution and has a sigmoid activation applied at the recurrent step and a tanh activation applied to its output. The number of units used at each LSTM layer varies with the number of layers used. In the first case, a CNN network regresses sun positions from a single RGB image with a resolution of 128 × 512. In the second row of Table 2, an input image's gradient map with magnitude and orientation is also considered. The gradient is computed by applying Sobel operators [42] vertically and horizontally for each RGB channel. The gradient maps are stacked, resulting in a 9-channel input (3 RGB image channels, three gradient magnitude channels, and three gradient orientation channels). Since the sun is responsible for the overall brightness distribution and accounts for sharp changes, image gradient maps can be used to guide the localization of the sun in images. Therefore, gradient maps were generated from input image sequences and provided as additional input to the CNN-LSTM network.

Table 1
Specifications of the CAU-2, Laval, and Sirta datasets.

Dataset		Laval	Sirta	CAU-2
Original resolution	Fisheye	1024 × 102	2160 × 3840	768 × 1024
	Equi-rectangular	512 × 2048	1327 × 4980	363 × 1452
Number of images used in training and testing		27,103	37,877	1711
Period (Years)		2023~2016	2017	2023
Sun Information	Elevation, Azimuth	Provided	Provided	N/A
	Irradiance	N/A	N/A	Provided

Table 2
Performance of the proposed method on the CAU dataset under two configurations (with and without gradient maps).

Input images	Models	MAE (°)	RMSE (°)	R^2 Score
Single	CNN	2.0150	3.4396	0.9390
	CNN with a gradient map	1.2465	2.6666	0.9630
Multiple	CNN-LSTM	0.9999	1.2765	0.9455
	CNN-LSTM with gradient maps	0.6263	0.7801	0.9973

In the second case, the proposed CNN-LSTM network has one LSTM layer and predicts sun positions from an image sequence with three frames at an inter-frame interval of 5 min. The last row of Table 2 shows the performance of a version of the CNN-LSTM modified to support gradient vector maps as additional inputs. The Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 score were computed from predicted and ground truth sun positions. MAE is computed as the sum of the absolute difference between predicted and ground truth data, and RMSE is their quadratic mean. The R^2 score measures how much an input variable accounts for the variance in predictions and has a range of 0–1. Higher R^2 scores imply a higher similarity between ground truth and predicted values. The equirectangular coordinate outputs of the model are converted to degrees using Bourke's approach [26] prior to loss computation. In Bourke's approach, normalized equirectangular image coordinates are transformed into 2D fisheye image coordinates represented as azimuth and elevation values. Since gradient maps represent the variations in brightness that occur as a result of the sun's presence in the sky, using them ensures that predicted sun positions are restricted to plausible regions of the sky. This is why the CNN with a single image and gradient map as input achieves a higher R^2 score than the CNN-LSTM with multiple input images, as shown in Table 2. Using the CNN-LSTM with multiple input images achieves better MAE and RMSE values because spatial and temporal features are both leveraged to achieve higher accuracy predictions. Table 2 shows that using gradient maps improves the performance of both the CNN and CNN-LSTM models. It can also be noted that using multiple frames offers significantly better performance than using a single image, whether or not gradient maps are used. The CNN-LSTM with gradient map inputs offers the best performance. Gradient maps were not used in other experiments because of the limited memory available.

In order to more accurately analyze the distribution of errors in Table 2, six whisker (box) plots were generated using the root of square distances between ground truth values and those predicted by three different networks (CNN, CNN-LSTM, and CNN-LSTM with gradient map inputs) as shown in Fig. 5. Only a single image was used for the CNN in Fig. 5 (a). Three frames at 5-min intervals were used for both CNN-LSTM networks to account for the brightness distribution features in each image as well as the temporal changes in them over consecutive frames. Image frames were resized to 128×512.

Only a single whisker plot is shown for the CNN network (a) because it has the largest variation in error values. Two pairs of plots ((b) and (c)) showing the same errors at different scales were generated for the two versions of the CNN-LSTM network. More specifically, the left plot for each of the two configurations ranges from 0 to 70° (same scale) to facilitate more straightforward comparison between error ranges. The right plot for Fig. 5 (b) and (c), however, ranges from 0 to the maximum error computed for each network's predictions (different scales) to provide a better highlight of their respective distributions. The maximum and minimum error values are represented by top and bottom horizontal strokes (whiskers). The box in each plot represents the first (bottom) as well as the third (top) quartiles, and the horizontal orange line in the box is the median error. Values outside the whiskers are considered to be extreme outliers. Fig. 5 (b) shows that using multiple frames instead of a single image as input drastically improves performance. Using gradient maps as additional inputs further improves performance, as shown in Fig. 5 (c).

The impact of the number of LSTM layers on the overall performance of the proposed method was examined. Model configurations with 1, 2, and 3 LSTM layers are shown in Fig. 6. Image sequences from the Sirta dataset with resolutions of 96×386 (8 frames) and 128×512 (3 frames) were used. Here, three frames were used for the input configuration with a larger resolution (128×512) because of the limited GPU memory available. Features are extracted from each input frame using a CNN model. The extracted features are passed to LSTM layers, which identify temporal changes in brightness distributions within them and predict sun positions. In Fig. 6,

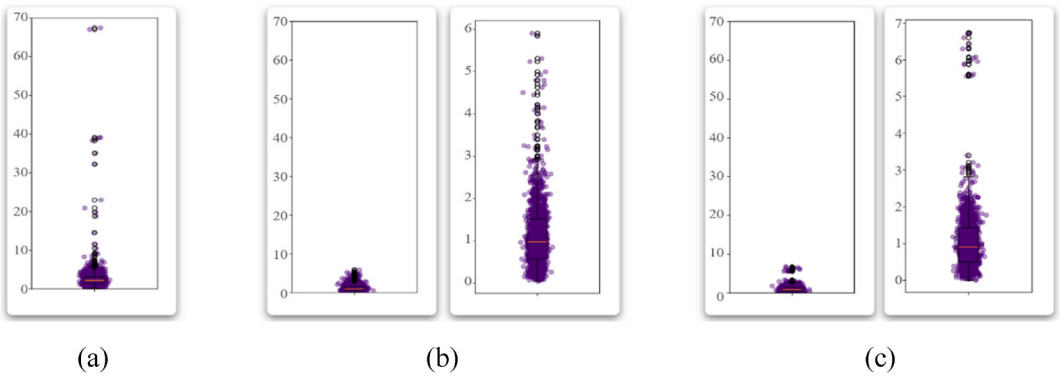


Fig. 5. Box plots generated from the root of squared distances between ground truth values and those predicted by three different networks: (a) CNN, (b) CNN-LSTM, and (c) CNN-LSTM with gradient map inputs.

Fig. 6. Structures of LSTM stacks used in experiments.

"LSTM 00" indicates that the number of LSTM units used in the LSTM layer is "00". Intermediate LSTMs have an orange border, indicating that a vector representation is returned for each time step. LSTM layers with no orange border return a single vector representation for all time steps. "Dropout 0.5" indicates that a dropout layer, intended to prevent overfitting, randomly excludes the representations generated by "50 %" of the nodes in the previous LSTM layer. "Dense 6" indicates that a fully connected layer regresses an elevation and azimuth prediction for each of the three input frames from the representations generated by the previous layer. For instance, a "Dense 16" layer is used when the network input is an image sequence with eight frames.

The results of the experiments performed are detailed in Table 3. Image sequences with an inter-frame interval of 5 min were generated from the Sirta dataset. Experiment results showed that the performance of the proposed method improves as the number of LSTM layers increases. Table 3 also shows that increasing the number of frames, even with lower-resolution images, improves the performance of models more than increasing the resolution of input images.

We conducted experiments to examine how the resolution and number of frames in an input image sequence affect the performance of the proposed network. Here, image sequences from the Sirta dataset with 3, 5, and 8 frames at 5-min inter-frame intervals and dimensions of 64 × 256, 96 × 384, and 128 × 512 were used, as shown in Table 4. The model configuration with three LSTM layers was used because it demonstrated the best performance in earlier experiments (Table 3). Experiment results show that performance improves as the number of frames increases. This is because temporal changes in spatial information from previous frames can be leveraged to ensure coherent predictions. More specifically, the LSTM layers are capable of identifying and leveraging the temporal

Table 3
Performance of the proposed method under model configurations with 1, 2, and 3 LSTM layers for image sequences with frames of two sizes (96 × 386 and 128 × 512).

Image Sequences	96 × 386 (8 Frames)			128 × 512 (3 Frames)		
No. of LSTM Layers	MAE (°)	RMSE (°)	R^2 Score	MAE (°)	RMSE (°)	R^2 Score
1	2.49806	3.85345	0.96582	2.57435	3.81025	0.96620
2	2.42364	4.04518	0.96242	2.46051	3.98191	0.96356
3	**2.21028**	**3.04331**	**0.97969**	2.86992	4.29837	0.95914

Table 4
Performance of the proposed model under various input configurations: image sequences with 3, 5, and 8 frames at three resolutions (64 × 128, 96 × 386, and 128 × 512).

No. of Frames	64 × 128		96 × 386		128 × 512	
	MAE (°)	RMSE (°)	MAE (°)	RMSE (°)	MAE (°)	RMSE (°)
3	6.72762	9.71191	2.77751	4.20234	2.72402	4.85665
5	4.53153	6.62348	2.70903	4.24188	2.69836	4.81317
8	2.96346	5.20833	**2.21028**	**3.04331**	2.21372	3.73632

changes in the brightness distributions of current and previous frames to estimate the position of the sun in the current frame, even when it is partly or entirely occluded. Additionally, training the proposed model with elliptical constraints ensures that predicted sun positions are consistent with the natural path of the sun.

The resolution of frames determines the precision with which the position of the sun is estimated. For instance, doubling frame resolution results in twice the number of samples over the same range of elevation and azimuth values. Experiment results show that increasing image resolution does not always improve model performance because larger image resolutions are associated with higher sensitivity to outliers (in weather conditions and sun positions), particularly with the RMSE measure. In order to address the sensitivity to outliers, more training data samples are required when higher-resolution images are used. Tables 3 and 4 demonstrate that, given the same dataset and hardware configuration, increasing the number of frames shall improve model performance more than increasing the image resolution.

To examine the impact of the ellipse consistency penalty (ECP) and elliptical shape penalty (ESP) on the performance of the proposed method, a CNN-LSTM model with one LSTM layer was trained with three different loss configurations (Table 5). Here, "Wing loss" is the primary loss term in all three configurations. As described in subsection 2.1, it is computed as an L1 loss for large values, but a log function is applied to small values. Each input image sequence from the Sirta dataset contains three frames (128 × 512) at 15-min intervals.

Additionally, a CNN-LSTM network was trained with and without the elliptical consistency and shape constraints. Fig. 7 shows the overall training and test losses computed at each epoch for both scenarios. As shown in Fig. 7, the test loss for the model trained without elliptical constraints is prone to random spikes even when its training loss drops consistently as training progresses. Experiment results in both Table 5 and Fig. 7 show that using the shape and consistency elliptical constraints, in addition to the primary loss function, offers the best performance.

In Table 6, the performance of two versions of the proposed method is compared to that of previous methods. In contrast to previous approaches where a single image was used as input, the proposed network predicts sun positions from an image sequence containing three frames at an inter-frame interval of 5 min. Here, "Sirta" indicates that the model was only trained on the Sirta dataset and evaluated with the same dataset. "Laval + Sirta" indicates that the model was first pre-trained using the Laval dataset and then fine-tuned on the Sirta dataset. These configurations were then evaluated on a segment of the Sirta dataset. In Sohag's method [4], an input image is converted to grayscale and blurred using a Gaussian filter before applying binary thresholding. Moments of the saliently bright regions detected by the binary threshold are then computed vertically and horizontally to identify sun positions. For these experiments, the method was re-implemented with adjustments made to account for distortions that may occur as a result of converting the fisheye images of the Sirta dataset into their equirectangular form. Under Sohag's approach, no values are provided under the "Laval + Sirta" section because no deep-learning networks were used, meaning pre-training was unnecessary.

As described in subsection 3.2, Hold-Geoffroy's approach [12] uses seven convolution layers and two fully connected layers to predict sun positions from a single input image. In our evaluations, the CNN network proposed was re-implemented, but only the sun position estimation head was retained. The sun's position was regressed directly rather than through bins to facilitate higher-resolution inputs. Additionally, the mean squared error was used in place of the Kullback–Leibler divergence used in Hold-Geoffroy's method [12]. Rahim's approach [22] was re-implemented for this evaluation. In Rahim's approach, the average values are computed across color channels, and an adaptive thresholding operation is performed. Cases in which no circle was detected were excluded from the final error computation. As shown in Table 6, the proposed method more accurately predicts sun positions than previous methods.

Fig. 8 shows estimated sun positions under various levels of cloud cover. The blue and red circles represent ground truth (Gt) and predicted (pred) sun positions, respectively. The squared distance (Dist) between predicted and ground truth sun positions is also indicated in degrees. The model configuration with three LSTM layers was used to predict sun positions from input image sequences with three frames at an inter-frame interval of 5 min. Each frame in the sequence had a resolution of 96 × 386. One of the more difficult cases is the 9th image (last column of the last row) of Fig. 8. The first row of Fig. 9 shows the two previous frames used to predict the position of the sun in this case. This implies that the last frame in the first row of Fig. 9 is the 9th image in Fig. 8. Although the sun is

Table 5
Performance of the proposed model under three loss configurations: combinations of the primary loss, ESP, and ECP constraints.

Loss Terms Used	MAE (°)	RMSE (°)	R^2 Score
Wing loss + ESP	2.7632	7.2623	0.9773
Wing loss + ECP	2.1504	6.2273	0.9821
Wing loss + ESP + ECP	**2.0977**	**6.2394**	**0.9805**

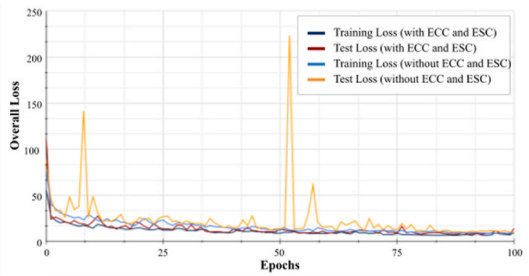


Fig. 7. Line plot showing the total training and test losses computed at various epochs during training for two configurations for the proposed CNN-LSTM network.

Table 6
Comparison of previous methods against two configurations of the proposed network.

No. of Input Images	Methods	Sirta			Laval + Sirta		
		MAE (°)	RMSE (°)	R^2 Score	MAE (°)	RMSE (°)	R^2 Score
Single	Sohag [4]	9.7480	24.8761	0.4656	–	–	–
	Hold-Geoffroy [12]	8.1071	10.9272	0.8655	7.1411	10.2836	0.8835
	Rahim [22]	6.7649	16.2021	0.6843	–	–	–
	Proposed (CNN)	3.7833	6.4054	0.9546	2.5509	6.2931	0.9541
Multiple	Proposed (CNN-LSTM)	2.4605	3.9819	0.9635	**2.1839**	**4.5687**	**0.9762**

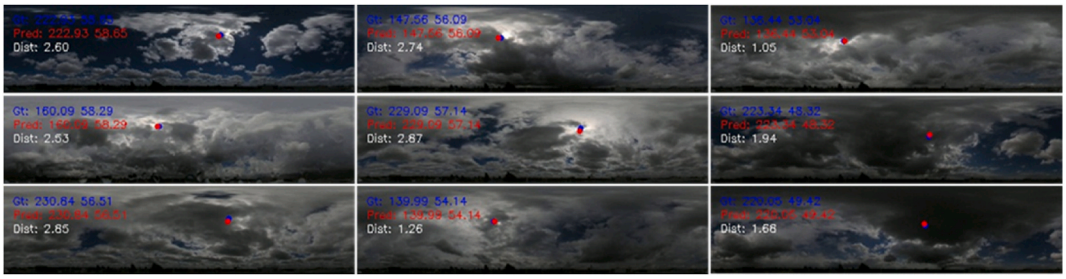


Fig. 8. Sun positions estimated using the proposed method under various levels of cloud cover.

entirely occluded by thick clouds in the current frame, the LSTM layers extract and leverage the temporal changes between features generated from previous frames. This is why the sun's position can be precisely estimated for this case for this and similar cases. In the second row of Fig. 9, the sun is also occluded in both the current and previous frames. Accurate sun position estimation is possible in this case because the proposed CNN-LSTM network is capable of adequately extracting the spatial features and the temporal changes in the brightness distribution necessary for sun position estimation, and the elliptical penalties ensure that predictions are consistent with the natural path of the sun.

The impact of the elliptical consistency and elliptical shape penalties on the performance of the proposed method was examined using scatter plots (Fig. 10). Here, a CNN-LSTM network with one LSTM layer was used to predict sun positions from image sequences (Laval dataset) with three frames, each with dimensions of 128 × 512. In Fig. 10 (a), ground truth points were plotted against those predicted by a CNN-LSTM network trained with only the primary loss term. Fig. 10 (b) shows predictions made using a CNN-LSTM network trained with both elliptical consistency and shape constraints. The red and green dots represent predicted and ground

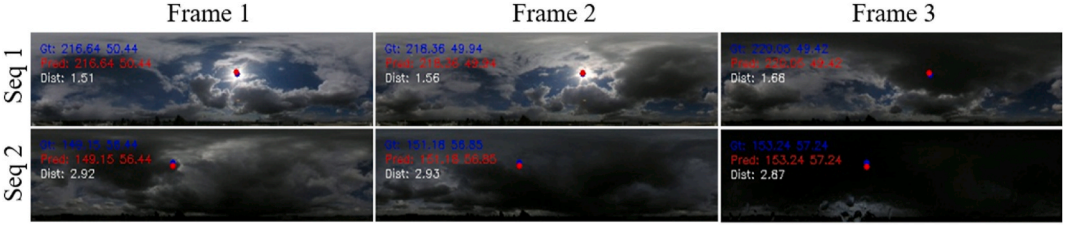


Fig. 9. Sun positions estimated from two image sequences, each with three frames.

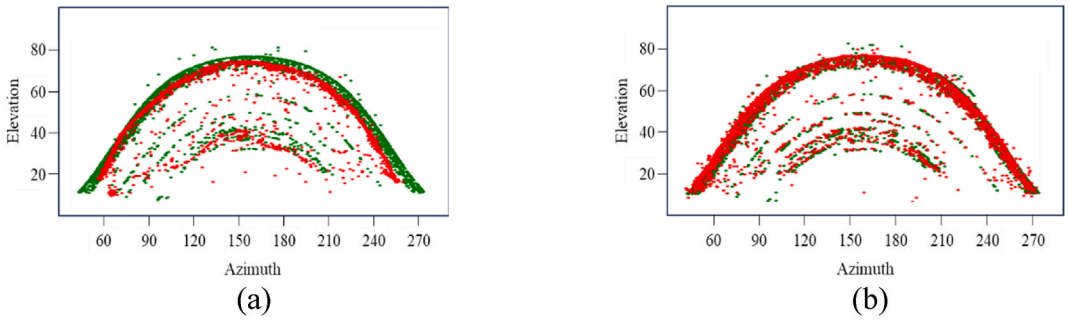


Fig. 10. Scatter plots of ground truth sun positions (green) as well as values (red) predicted by two configurations of the CNN-LSTM network. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

truth sun positions. As shown in Fig. 10 (b), using the constraints based on elliptical geometric properties can ensure that predictions are plausible, even when clouds partly or entirely occlude the sun. The predictions shown in Fig. 10 (b) are more consistent with the sun's natural path than those shown in Fig. 10 (a).

In order to compare the inference time of the proposed approach against that of previous approaches, 4031 images from the Sirta dataset were resized to dimensions of 128 × 512 and fed as input to each of the methods listed in Table 7. In order to consider the temporal changes in the brightness distribution over multiple frames, the CNN-LSTM networks are fed multiple (three) image frames as input. The average execution time in milliseconds is computed over all the input images. It can be noted that for a single frame, the two deep learning approaches, Hold-Geoffroy [12] and the proposed CNN, offer significantly better performance than Sohag [4] and Rahim's [22] non-deep learning methods while maintaining fast inference times of under 4 ms. In order to consider the temporal changes in the brightness distribution over multiple frames, the CNN-LSTM networks are fed multiple (three) image frames as input. The proposed CNN has slightly longer inference times than Hold-Geoffroy's approach because it has significantly more convolution layers that allow it to improve accuracy. The CNN-LSTM and CNN-LSTM with gradient map inputs have longer inference times because they process three times as much data. This implies that the inference time per frame is 7.83 ms and 7.86 ms for the CNN-LSTM and CNN-LSTM with gradient map, respectively. Since the elliptical constraints are only employed during training, using them does not increase inference time.

5. Discussion

In the proposed approach, spatial, temporal, and geometric features are considered while regressing sun positions from image sequences. More specifically, a CNN-LSTM model extracts spatial features and leverages temporal changes in the brightness distribution across sky image sequences to regress sun positions. Also, elliptical (geometric) constraints are added to the loss computation, ensuring that sun position predictions are consistent with the sun's natural path.

Table 2 shows that leveraging temporal features in addition to spatial features achieves significantly better performance than only using the spatial features of a single frame. Table 3 shows that increasing the number of LSTM layers and the number of frames, even with lower-resolution images, improves the performance of models. However, because of the limited processing power available, experiments were only conducted with up to three LSTM layers. Table 4 demonstrates that increasing the number of frames improves model performance more than increasing the image resolution.

The improvement in performance is associated with two key factors. First, changes in spatial image features over more frames are sufficiently considered. This implies that considering a more extensive temporal context improves performance. Secondly, more sun positions are used in computing and enforcing the elliptical constraints, resulting in more plausible predictions. More specifically, increasing the number of frames that support the estimated elliptical equation improves overall performance. Table 5 shows the performance of the proposed model under three loss configurations: combinations of the primary loss, elliptical consistency constraint (ECC), and elliptical shape constraints (ESC). The ECC ensures that the properties of the ellipse of best fit through a sequence of sun positions are consistent with those occurring in ground truth data. However, with the ECC alone, it is possible to have a plausible ellipse of best fit with no points lying on it. The ESC ensures that predicted sun positions lie on their ellipse of best fit. Table 5 shows that considering both the elliptical consistency and elliptical shape constraints achieves the best performance.

In Table 6, the performance of the proposed method is compared to that of previous methods in which a single image was used as input. Although Hold Geoffroy's convolutional approach outperforms the algorithmic methods proposed by Rahim [22] and Sohag [4], the single image-based version of the proposed approach achieves better performance, as shown in Table 6. The proposed CNN-LSTM achieved the best performance because it considers spatial as well as temporal features in identifying the position of the sun. In addition, Paletta's approach [9] leverages multiple observations to estimate the position of an occluded sun. This is achieved by performing interpolation over past observations taken at the same time on previous days and interpolating through observations made from the start of the current day. A major limitation of this approach is that it requires that sky images be captured at a single location at regular intervals over an extended period of time, resulting in increased setup time and memory requirements. This makes it unsuitable for dynamic use cases that may require the rig to be moved or reoriented in contrast to the proposed approach. An error of 4.7

Table 7
Comparison of the proposed method's inference times in milliseconds (ms) against those of previous approaches.

No. of Input Images	Single				Multiple	
Method	Sohag [4]	Rahim [22]	Hold -Geoffroy [12]	Proposed CNN	Proposed CNN-LSTM	Proposed CNN-LSTM (Grad)
Sun Position Estimation Time (ms)	0.80	4.50	2.74	3.64	23.50	27.22

pixels (0.91 % of the image size) was reported in Ref. [9]. The mean average error in degrees of the proposed CNN-LSTM, shown in Table 6, was recomputed in pixels to 4.5 pixels (0.88 % of the image size).

Although the proposed sun position estimation approach accurately identifies the sun's position under a wide range of weather and illumination conditions, some challenges still need to be addressed, as highlighted below.

In sky imaging, Neutral Density (ND) filters are often used to reduce the intensity of light incident on camera lenses. However, an inappropriate ND filter level may result in highly overexposed images of clear skies or under-exposed images of heavy cloud cover. In cases such as these, where pixel values do not sufficiently represent the brightness distribution of the sun, image-based methods, including the proposed approach, are unable to make plausible predictions. Therefore, in future works, an exposure selection module that ensures a rich brightness distribution in input frames is to be implemented. In order to recover brightness distributions from under or overexposed images, we also intend to explore exposure correction techniques.

The CNN-LSTM architecture used in the proposed approach requires that a fixed number of frames is used for training and inference. In order to support longer context lengths with a variable number or size of image frames, the proposed approach requires modifications to its architecture.

In order to achieve robust sun position estimation across a range of scenes, models need to be trained using multiple large datasets. However, the computational requirements of the CNN-LSTM architecture used in the proposed approach limit its ability to scale to extremely large datasets. Here, an attention-based architecture, such as the vision transformer [43] could be employed in place of the CNN components.

The proposed CNN-LSTM network, which leverages spatial, temporal, and geometric features is capable of regressing sun positions from a sequence of images even when the sun is partly or entirely occluded. It can, therefore, be leveraged to design and develop more accurate solar energy forecasting systems. The proposed omnidirectional imaging system comprises a commercially available camera, fish-eye lens, and panoramic tripod head. This implies that it could be adapted to create portable lighting rigs for augmented reality applications.

6. Conclusion

This study proposed a real-time sky imaging system and a CNN-LSTM network that regresses sun positions from an input image sequence. The proposed CNN-LSTM network is capable of reliably estimating the position of the sun because it considers both spatial and temporal features. In the proposed approach, spatial features are extracted from each frame in the input image sequence using a Resnet-based CNN architecture. In order to consider the temporal changes in the brightness distribution over multiple frames, the output of the CNN networks is concatenated and passed to a stack of LSTM layers. In addition to the primary loss term computed as the MAE between predicted and target sun positions, elliptical shape and consistency constraints are included in the training loss computation to ensure that the sun positions predicted by our CNN-LSTM network are consistent with the sun's natural path. In other words, training the proposed CNN-LSTM network with elliptical constraints minimizes the abrupt variations in estimations that may occur as a result of extremely heavy cloud cover. In order to evaluate the performance of the proposed approach, experiments were conducted using the Sirta and Laval datasets as well as a custom dataset (CAU dataset). The proposed approach achieved an R^2 Score of 0.98 on the CAU dataset. This is at least 0.1 higher than previous approaches. Because the proposed approach only consists of a single commercially available omnidirectional camera with a fish-eye lens and a panoramic tripod head, it is ideal for dynamic outdoor applications where the position and orientation of rigs may be frequently altered. These include light source estimation for solar power and augmented reality systems, as well as localized weather forecasting and building orientation optimization. In order to facilitate robust training with multiple large datasets, attention-based architectures such as the vision transformer could be employed in place of the computationally intensive CNN and LSTM components used in the proposed approach. In order to facilitate robust training with multiple large datasets, attention-based architectures such as the vision transformer could be employed in place of the computationally intensive CNN and LSTM components used in the proposed approach. In order to address the under or over-exposure challenges associated with using a fixed ND-Filter value, we intend to implement a network that dynamically identifies the most appropriate exposure for a given scene. In order to recover brightness distributions from under or over-exposed sky images, exposure correction techniques that leverage the relationship between frames shall also be explored.

Availability of data and materials

The implementation of the proposed method and the CAU-2 dataset are available at https://github.com/markompab/CNN-LSTM-Models-using-Elliptical-Constraints-for-Temporally-Consistent-Sun-Position-Estimation.

Funding statement

This research was supported by the Chung-Ang University Young Scientist Scholarship in 2022.

CRediT authorship contribution statement

Mark Mpabulungi: Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Kyeongmin Yu:** Software, Methodology, Data curation. **Hyunki Hong:** Writing – review & editing, Writing – original draft, Supervision, Software, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors extend their appreciation to the team at the Sirta Atmospheric Observatory and Professor Jean-Francois Lalonde and his team for their excellent work on the Sirta and Laval sky mage datasets, respectively.

References

[1] C. Yinghao, L. Mengying, C.F. Carlos, Sun-tracking imaging system for intra-hour DNI forecasts 96 (Part A) (2016) 792–799.
[2] L. Barthès, M. Haeffelin, O. Bock, C. Boitel, S. Bony, D. Bouniol, H. Chepfer, C. Marjolaine, J. Cuesta, J. Delanoë, SIRTA, a ground-based atmospheric observatory for cloud and aerosol research, Ann. Geophys. 23 (2) (2005) 253–275.
[3] T. Zhu, H. Wei, X. Zhao, C. Zhang, K. Zhang, Clear-sky model for wavelet forecast of direct normal irradiance, Renew. Energy 104 (2017) 1–8.
[4] H.A. Sohag, M.a.K.M. Hasan, M. Ahmad, An accurate and efficient solar tracking system using image processing and LDR sensor, in: *Proceedings of the International Conference on Electrical Information and Communication Technologies (EICT)*, Khulna, Bangladesh, 2015.
[5] B. Philippe, W. Lucien, The SG2 algorithm for a fast and accurate computation of the position of the Sun for multi-decadal time period, Sol. Energy 86 (10) (2012) 3072–3083.
[6] B. Adam, S.R. West, D. Rowe, S. Saad, A. Berry, Short-term irradiance forecasting using skycams: motivation and development, Sol. Energy 110 (2014) 188–207.
[7] F. Cong, Z. Jie, Z. Wengi, H. Bri-Mathias, 15 March, Convolutional Neural Networks for Intra-hour Solar Forecasting Based on Sky Image Sequences, vol. 310, 2022 118438.
[8] T.A. Siddiqui, B. Samarth, K. Shivkumar, A deep learning approach to solar-irradiance forecasting in sky-videos, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Waikoloa Village, HI, 2019.
[9] P. Quentin, L. Joan, A Temporally consistent image-based sun tracking algorithm for solar energy forecasting applications, in: NeurIPS 2020 - Tackling Climate Change with Machine Learning Workshop, 2020. Vancouver.
[10] X. Zhao, W. Haikun, W. Hai, Z. Tingting, Z. Kanjian, 3D-CNN-based feature extraction of ground-based cloud images for direct normal irradiance prediction, Sol. Energy 181 (2019) 510–518.
[11] R.H. Inman, H.T. Pedro, C.F. Combria, Solar forecasting methods for renewable energy integration, Prog. Energy Combust. Sci. 39 (2013) 535–576.
[12] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, J.-F. Lalonde, Deep outdoor illumination estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.
[13] L. Hošek, A. Wilkie, An analytic model for full spectral sky-dome radiance, ACM Trans. Graph. 31 (2012).
[14] H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian, Identity mappings in deep residual networks, in: Proceedings of the European Conference on Computer Vision, 2016. Amsterdam.
[15] J.-F. Lalonde, L.-P. Asselin, J. Becirovski, H.-G. Yannick, M. Garon, M.-A. Gardner, J. Zhang, The Laval HDR Sky Database, 2016 [Online]. Available: http://sky.hdrdb.com/.
[16] R. Walraven, Calculating the position of the sun, Sol. Energy 20 (1978) 393–397.
[17] J.J. Michalsky, The astronomical almanac's algorithm for approximate solar position (1950–2050), Sol. Energy 40 (3) (1988) 227–235.
[18] J.H. Meeus, Astronomical Algorithms, Willman-Bell Incorporated, Richmond, 1991.
[19] R. Grena, An algorithm for the computation of the solar position, Sol. Energy 82 (2008) 462–470.
[20] J.-F. Lalonde, A. Efros, N. Srinivasa, Estimating the natural illumination conditions from a single outdoor image, Int. J. Comput. Vis. 98 (2012) 123–145.
[21] R. Perez, R. Seals, J. Michalsky, All-weather model for sky luminance distribution—preliminary configuration and validation, Sol. Energy 50 (3) (1993) 235–245.
[22] A.R. Rahim, M. Zainudin, M. Ismail, M. Othman, Image-based solar tracker using Raspberry Pi, Journal of Multidisciplinary Engineering Science and Technology (JMEST) 1 (5) (2014).
[23] X. Chen, X. Jin, K. Wang, Lighting virtual objects in a single image via coarse scene understanding, Sci. China Inf. Sci. 57 (2014) 1–14.
[24] L. Tangwei, T. Guanjan, T. Hongying, L. Baoqing, C. Bo, Fisheyedet: a self-study and contour-based object detector in fisheye images, IEEE Access 8 (2020) 1739–71751.
[25] P. Goodarzi, M. Stellmacher, M. Paetzold, A. Hussein, E. Matthes, Optimization of a cnn-based object detector for fisheye cameras, in: Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2019. Cairo.
[26] P. Bourke, dualfish2sphere [Online]. Available: http://paulbourke.net/dome/dualfish2sphere/, , 2016, 27 20 07 2023.
[27] H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016.
[28] Q. Xiao, X. Chang, X. Zhang, X. Liu, Multi-information spatial-temporal LSTM fusion continuous sign language neural machine translation, IEEE Access 8 (2020) 216718–216728.
[29] C. Su, H. Huang, S. Shi, P. Jian, X. Shi, Neural machine translation with Gumbel Tree-LSTM based encoder, J. Vis. Commun. Image Represent. 71 (2020) 102811.
[30] Y. Korkmaz, A. Boyacı, Hybrid voice activity detection system based on LSTM and auditory speech features, Biomed. Signal Process Control 80 (2023) 104408.
[31] S.-Y. Chang, B. Li, G. Simko, T.N. Sainath, A. Tripathi, A. van den Oord, O. Vinyals, Temporal modeling using dilated convolution and gating for voice-activity-detection, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2018. Calgary.
[32] Y. Rahmi, R.F. Mohommad, M. Muliadi, I. Fatma, A. Friska, B. Irwan, E.P. Septyan, LSTM and Bi-LSTM models for identifying natural disasters reports from social media, Journal of Electronics Electromedical Engineering and Medical Informatics 5 (4) (2023) 241–248.

[33] S. Hochreiter, S. Jürgen, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.
[34] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: a search space odyssey, Transactions on neural networks and learning systems 28 (10) (2016) 2222–2232.
[35] F. Zhen-Hua, K. Josef, A. Muhammad, H. Patrik, W. Xiao-Jun, Wing loss for robust facial landmark localisation with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. Salt Lake City.
[36] H. Radim, J. Flusser, Numerically stable direct least squares fitting of ellipses, in: Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualisation, 1998.
[37] D.F. Rogers, A.J. Adams, in: Mathematical Elements for Computer Graphics, McGraw Gill, 1990, p. 75.
[38] Newton-raphson method, in: B. Engquist (Ed.), Encyclopedia of Applied and Computational Mathematics, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 1023–1028.
[39] D. Hasler, S.E. Suesstrunk, Measuring colourfulness in natural images, Human vision and electronic imaging VIII 5007 (2003) 87–95. SPIE.
[40] International Electrotechnical Commission and others, Multimedia systems and equipment-color measurement and management-Part 2-1, Color management-Default RGB color space-sRGB (1999).
[41] J. Kannala, S.S. Brandt, A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1335–1340.
[42] N. Kanopoulos, N. Vasanthavada, R.L. Baker, Design of an image edge detection filter using the Sobel operator, IEEE J. Solid State Circ. 23 (2) (1988) 358–367.
[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, J. Llion, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017) 5998–6008.