Cloud detection using convolutional neural networks on remote sensing images

Lysha M. Matsunobu, Hugo T.C. Pedro, Carlos F.M. Coimbra *
Department of Mechanical and Aerospace Engineering, Jacobs School of Engineering, Center of Excellence in Renewable Resource Integration and Center for Energy Research, University of California, San Diego, La Jolla, CA 92093, USA

ABSTRACT

Cloud detection is an important task for remote sensing and solar resource modeling, and an initial step toward more complex tasks like solar forecasting. Recent advances in remote sensing have increased the spectral, spatial, and temporal resolution of observations. This study demonstrates the image analysis capacity of convolutional neural networks (CNNs) to identify the presence of clouds from remote sensing directly, without ancillary data and at relatively high temporal resolution. Cloud detection in images from the Geostationary Operational Environmental Satellite (GOES)-16 Advanced Baseline Imager (ABI) is validated against ground telemetry from a set of 12 locations of diverse geographic and climatic conditions across the continental U.S. (CONUS). As spatial coverage through ground station networks is typically sparse, transfer learning of baseline models is studied in order to assess the robustness and portability of baseline models through transfer learning scenarios. Performance of the CNN-based cloud mask (CCM) is discussed in three parts: (1) baseline training in comparison to a deterministic model, the ABI cloud mask (ACM), (2) transfer learning of baseline models, and (3) transfer learning of models trained on combined location pairs. In baseline models, the CCM showed average percent improvement over the ACM of 11% in accuracy (ACC) and 30% in Matthews correlation coefficient (MCC) score. In transfer learning scenarios, adding a second location to training samples improved aggregate performance by 18.8%. Results show that CCM transfer learning is highly asymmetric. A framework is proposed to characterize and predict transfer learning performance with respect to this asymmetry.

1. Introduction

Solar energy is a well-established and rapidly growing provider of carbon-free energy globally (Zhang et al., 2015a; Voyant et al., 2017; IEA, 2021). However, the variability of solar resource and its strong dependence on a suite of environmental parameters (e.g. cloud cover, aerosols, geographic topology, surface temperature) require accurate solar resource assessment and forecasting models for further adoption at high penetration levels (Inman et al., 2013; Yang et al., 2018). Toward this aim, data sets to build and validate solar irradiance models are critical to solar energy integration (Pedro et al., 2019). Recent advances in remote sensing offer new opportunity to improve solar irradiance models (Larson et al., 2020), both for historical assessment and implementation in real-time operations, and provide observations at a broad spatial coverage that would be nearly impossible to achieve with ground monitoring networks.

The determination of cloud state is critical to remote sensing data products. Remote sensing of atmospheric conditions like aerosol loading and surface conditions like solar irradiance and vegetation mapping are strongly influenced by cloud optical properties (Ishida and Nakajima, 2009; Larson et al., 2020). Cloud masking is an intermediate step in establishing modeling regimes. Because data products in remote sensing are strongly influenced by cloud optical properties, cloud masking can be used to differentiate areas of clear and cloudy skies. Cloud masking directly from next-generation remote sensing at relatively high resolution (5-minute) is an important step for the use of remote sensing observations as proxies for ground observations of solar irradiance, and it is also an important initial step in demonstrating the feasibility of new methods for improving solar forecasting based on remote sensing.

In this study, a purely statistical machine learning model is developed and validated for the Advanced Baseline Imager (ABI) on the latest Geostationary Operational Environmental Satellite (GOES) series, GOES-R (Schmit et al., 2017). A convolutional neural network (CNN)-based cloud mask (CCM) model is shown to be competitive with purely physical models like the ABI cloud mask (ACM) across a range of geographic and climatic locations in the continental U.S. (CONUS).
Discussion of data retrieval, processing, and input characterization is included in Section 2. CNN model structure is outlined in Section 3, with further detail given in Appendix A. CNN performance and results are discussed in Section 4. Supplementary definitions and discussion of binary classification metrics are provided in Appendix B.

1.1. Cloud masks

Cloud masks, and the related task of cloud detection, are critical to remote sensing. The presence of clouds has strong effects on radiation data, aerosol retrievals, and observation of surface conditions like vegetation (Ishida and Nakajima, 2009). Cloud masks may be employed to remove cloud contamination in surface temperature estimation (Ishida et al., 2018) and aerosol retrievals (Lyapustin et al., 2008), pre- or post-process solar forecasting output and radiative transfer models (RTM) (Inman et al., 2013; Li et al., 2017; Miller et al., 2018), and initialize the cloud state in NWP models (Jimenez, 2020).

Cloud masks may report a binary or multi-class determination of cloudy states. Though the ACM officially produces a binary mask, an intermediate product is the four-level, NASA and NOAA heritage mask with the categories: clear, probably-clear, probably-cloudy, and cloudy. The ACM uses 9 of the 16 ABI channels and labels each pixel as clear or cloudy according to threshold tests from primary sensor data (ABI observations) and exogenous data like ancillary masks, numerical weather prediction (NWP) forecasts, and RTM data. Following determination of the four-level mask, pixels labeled probably-clear or clear are classified clear in the final binary mask. The full algorithm is outlined in detail in the ACM technical document (Heidinger and Straka III, 2013). Favorable performance of the CCm in comparison to a well-calibrated, deterministic model like the ACM demonstrates the clear potential for use of CNN methods directly on remote sensing data and the potential for real-time use noting that a CCM operates without ancillary data dependencies. One assessment of the ACM under all-sky conditions found that accuracy is approximately 90.9% across CONUS, noting that performance degrades for locations north of parallel 36 (Jimenez, 2020).

The ACM includes both day and night pixels, with optional additional threshold tests performed on daytime pixels (Heidinger and Straka III, 2013). Because daytime cloud masks are of greater interest for solar forecasting applications and since target labels were derived from GHI data, this study only considers performance of the cloud mask during daytime.

1.2. Machine learning for cloud masking

CNNs present enormous potential for image classification and within solar energy applications, they have been applied to cloud images, typically collected from total sky imagers (TSI). In one such study, a CNN was developed to predict 15-minute ahead photovoltaic (PV) power output at 1-minute resolution. As a nowcasting model, this CNN achieved 26.0% to 30.2% relative root mean square error (rRMSE) and found inconsistent model performance with increasing architectural complexity beyond two convolutional layers and two fully connected layers (Sun et al., 2018). As a forecasting model, it achieved 15.7% forecast skill in all weather conditions through a hybrid input model, including lagged sky images and PV power output data (Sun et al., 2019); then later 17.1% forecast skill relative to persistence after exploration of input and output feature assemblies (Venugopal et al., 2019).

Other recent applications of CNNs to sky images have the objective of providing high-resolution cloud coverage information. In one case, researchers sought to obtain detailed binary cloud masks from all-sky images (Xie et al., 2020). In another case, CNN models were trained to identify not just the presence of clouds but also the cloud type (Fabel et al., 2021). In both models, the CNNs surpassed the classification accuracy of classic segmentation techniques that rely on threshold-based methods in the color space.

More broadly, other machine learning methods have been combined with remote telemetry to improve classification tasks and forecasting models. Naive Bayesian (Heidinger et al., 2012), support vector machine (Ishida et al., 2018), and Markov random field (Hégarat-Mascle and André, 2009) methods have been used for cloud detection with respect to NOAA’s Advanced Very High Resolution Radiometer (AVHRR), Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO), and Moderate Resolution Imaging Spectro-radiometer (MODIS) data. One study used random forests to explore the predictor importance and multicollinearity of GOES-16 channels and High-Resolution Rapid Refresh (HRRR)-modeled meteorological variables in producing a binary cloud mask (McCandless and Jimenez, 2020). In more recent studies, CNNs have started to be used for cloud detection and semantic segmentation and show potential improvement over existing detection algorithms (Luotamo et al., 2020; Mateo-García et al., 2017). CNNs have recently been employed for solar irradiance prediction, using cloud masks for initialization (Pérez et al., 2021).

Few studies have applied CNNs to GOES telemetry from the updated ABI for either cloud detection or forecasting. This state of affairs will change rapidly given that the new ABI sensor offers approximately three times spectral resolution, four times spatial resolution, and five times temporal resolution compared to the previous imager (Kalluri et al., 2018). In one of those studies (Cintineo et al., 2020), CNNs ingested ABI and flash-extent density from the Geostationary Lightning Mapper (GLM) to extract relevant features and patterns in the data for the prediction of intense convection associated with threatening thunderstorms. Another study (Hilburn et al., 2020) applied CNNs to ABI and GLM data to produce synthetic radar reflectivity fields that can be assimilated by NWP models.

1.2.1. Transfer learning with remote sensing data

Transfer learning, a rapidly growing area of machine learning, is applied under the assumption that knowledge can be transferred to target domains that are similar to source domains; analogous to a person’s ability to generalize beyond personal experience (Zhuang et al., 2021). The method of transfer learning used in this study can be classified as transductive transfer learning, sometimes called domain adaptation, where source labels are available, target labels are not, and the input features are the same but their marginal distributions may be different (Pan and Yang, 2010; Goodfellow et al., 2016; Zhuang et al., 2021). The input features are equivalent across source locations, in input structure and processing methodology, however, the target class probabilities are unique to each location and average spectral signals may vary according to typical meteorological patterns or surface conditions.

Some studies have shown the promise of transfer learning in adapting cloud masks across remote sensing multispectral sensors. In two such studies, CNNs are trained on cloud masks from Landsat 8 images then applied to images from Proba-V (Mateo-García and Gómez-Chova, 2018) and Sentinel-2 (López-Puigdollers et al., 2021). In particular, the performance of transfer learning schemes was shown to have strong dependence on the particular train and test data set used (López-Puigdollers et al., 2021), but further characterization of this dependence was not explored. To the best of the authors knowledge, this is the first study to apply CNN transfer learning to GOES-16 ABI data. In addition, this study uses a baseline CNN trained on ground station telemetry, instead of existing rule-based or manually-labeled cloud masks, to better facilitate use of a CCM in assessing ground conditions.

1.3. Choice of binary classification metric

In binary classification, the most representative evaluation of performance is the classification matrix, which identifies true and false targets versus predictions, comparing model predictions against the
target classification set. There are a number of metrics that employ portions of the classification matrix to assess performance, notably recall or true positive rate (TPR), precision or positive predictive value (PPV), accuracy (ACC), F1 score (F1), and the Matthews correlation coefficient (MCC). Simpler metrics exhibit biased behavior when the classes are significantly unbalanced, whereas MCC score is notable as a robust metric to capture both model accuracy and classification error (Chicco and Jurman, 2020; Brown, 2018; Sokolova and Lapalme, 2019; Luque et al., 2019). Further discussion is included in Appendix B. Many studies report accuracy for cloud and clear sky masks so accuracy scores are provided for comparison with other studies, however model training and performance in this study were evaluated primarily using MCC scores, which capture both clear and cloudy classification in a single metric without biasing a dominant, majority class.

2. Data

Data was collected for the period January 1, 2018 to December 31, 2018 from remote sensing and ground telemetry. Data and processing methods are discussed in the following sections.

2.1. Remote telemetry

Spectral radiation data was collected from GOES-R, officially GOES-East, from all 16 available channels of the ABI instrument (GOES-R Calibration Working Group, 2017). Of these channels, 6 are in the visible and near-infrared and 10 are in the thermal infrared (Kalluri et al., 2018). Images are retrieved from CONUS scans, which have 5-minute temporal resolution (Schmit et al., 2017). The 16 channels are interpolated to equivalent 11 by 11 pixel grids of 0.5 km per pixel resolution, the finest resolution among ABI channels, centered on locations with known ground stations from the Surface Radiation Budget Network (SURFRAD) and Solar Radiation (SOLRAD) network. In addition to spectral radiation data, the binary cloud mask data product of the ABI (GOES-R Algorithm Working Group and GOES-R Series Program, 2018) was retrieved to provide a point of comparison for the CCM binary classification algorithm. The ACM officially provides a binary classification per pixel. Mask results were matched to input data based on closest pixel to location of corresponding station. Quality flags may be assigned in cases where view angle is outside of acceptable range or key ABI sensor data is missing or determined to be of poor quality (Heidinger and Straka III, 2013). All time stamps with data quality flags were removed when evaluating the ACM performance, which significantly reduced available samples. When CCM scores are directly compared with ACM scores, they are matched by timestamp so that differences in sample size and class size do not hinder fair comparison. The number of samples used to evaluate performance of the CCM alone is given by the test column, whereas the number of samples evaluated when the CCM is directly compared to the ACM is given by the comparison column in Table 1.

Image normalization by channel range and solar zenith angle \(\theta_v\) was employed for the benefit of CNN training, in order to equally weight each ABI channel image and to remove diurnal effects. Each ABI channel has a minimum and maximum spectral response value as outlined by Padula and Caio (2015) for shortwave channels and Hillger and Schmit (2004) for longwave channels. There are 16 ABI channels \((l)\) for which the minimum and maximum value per channel can be represented, respectively, as \(\min(L_l)\) and \(\max(L_l)\). The normalization method for pixels in each ABI image can be expressed as

\[
\hat{L}_{\text{norm}} = \frac{L_{\text{norm}} - \min(L_l)}{\max(L_l) - \min(L_l)},
\]

(1)

where \(L_{\text{norm}}\) is the initial value of the image pixel after mapping all 16 channels to the same grid and \(L_{\text{norm}}\) is the resultant normalized pixel value. Here, \(m, n \in [1, ..., 11]\) and \(l \in [1, ..., 16]\).

Some resultant pixel values may be negative, as division by a small \(\cos(\theta_v)\) could result in a term greater than 1 subtracted from 1. It should be expected that no normalized pixel values are much greater than 1 as that would imply a negative value in the numerator and an error with channel bounds.

Prior to training the CNN, images from a training set are zero-centered. In this process, a per channel mean is determined from training set samples, then subtracted from all pixel values in the training, validation, and test set, such that the distribution is forced toward a zero mean. This process is analogous to batch normalization to prevent covariate shift. Mainly, this ensures that slight differences in the average value of training sets versus validation or test sets do not have outsize impact on performance.

2.2. Ground telemetry

Twelve stations from the SURFRAD (Augustine et al., 2000) and SOLRAD (Hicks et al., 1996) networks were selected to represent a diversity of geographic and climate conditions. Station location and elevation are provided in Table 1. GHI observations from each station were retrieved for 2018 at 1-minute temporal resolution.

Quality control of radiation data in these two networks follows the algorithms implemented for the Baseline Surface Radiation Network (Long and Shi, 2008). Bad data are deleted, but questionable data are only flagged. In this work, all flagged GHI values and any unrealistic values (i.e. negative GHI) were removed from analysis. The GHI data was downsampled to 5-minute resolution using backward window averaging, such that ground data could be aligned with ABI CONUS scans. Nighttime values, with solar zenith angles greater than 85°, were removed. The open-source pvlib package (v.0.8.1) was used to find solar zenith angle and clear sky GHI values (Holmgren et al., 2018). The pvlib algorithm, detect_clearsky(), was employed to assess clear sky classification. The algorithm is based on the Reno and Hansen clear sky model (Reno and Hansen, 2016), which calculates a series of statistics to compare to threshold values in order to determine the sky condition. A sliding window of 20 min, to accommodate the 5-minute resolution, was employed for clear sky detection. While other notable clear sky detection methods exist (Inman et al., 2015; Bright et al., 2020), the Reno and Hansen model is used as a standard model in the industry (Yang et al., 2020) and provides good performance of clear sky detection on time series of GHI data alone at resolutions up to 15-minutes (Reno and Hansen, 2016).

Previous reports have shown the ACM has a clear sky identification accuracy of 85% to 90% when validated against CALIPSO retrievals (Heidinger and Straka III, 2013; Jimenez, 2020; McCandless and Jimenez, 2020). In this study of 12 CONUS locations, where clear sky ground truth is determined using the ACM and Hansen (Reno and Hansen, 2016) model, the average ACM accuracy is 80.9%. This lower accuracy is likely the result of using ground versus remote sensing telemetry for validation. The ACM represents an area the size of the given pixel, up to 2 km for the CONUS data product (Heidinger and Straka III, 2013); whereas the clear sky state determined by ground telemetry is determined for a specific point (Palmer et al., 2018). It has also been shown that satellite cloud classification tends to overestimate clear skies in a comparison of the Reno and Hansen model to the previous generation of GOES satellite (Reno and Hansen, 2016). This tendency is supported by results of this study where the ACM has a higher false positive rate in identifying clear skies than the CCM, Fig. 1. For this study, the determination of model accuracy is a fair comparison since the ACM and CCM are both subject to cloud state as defined by ground telemetry.

2.3. Partition

After collecting and preprocessing remote and ground telemetry, the respective sets of 5-minute resolution data were joined on timestamps. A list of unique dates was extracted from the joined data set and dates were randomly partitioned such that training represents 80% of
the data and testing is 20% of the data. The training set was further split into training and validation as 85% to 15%, respectively. Sets were partitioned by dates since splitting on timestamps could serve to advantage a model that was trained on a sample only 5-minutes apart from a test set sample (Sun et al., 2019). Partitioning on dates, such that data from a single day is always assigned to a single set, should allow more realistic performance for an operational model. Random partitioning was completed using the train_test_split() function from scikit-learn (v.0.23.2) with a random state instance applied to ensure reproducible shuffling. Model tuning was completed using only the training and validation sets. Final performance metrics are reported based on a set of 10 iterations of model training. The number of samples included in each data set per location are listed in Table 1.

2.4. Input channel characteristics

In later analysis, discussed in Section 4.2.1, performance shows correlation with the average spectral response and distribution values of certain ABI channels. To quantify this association, the following sampling process was used to retrieve representative values of channel average and variance per location. At even spacing through each location’s training set, 10,000 sample timestamps were retrieved. Sampling occurred on processed data, meaning after each pixel value was normalized by channel range and solar zenith angle, but before each sample had been zero-centered to adjust for covariate shift. The result, for each location, was an array of size [10^7, 16, 11, 11]. Next, each channel image was reduced to its average value, or size [10^4, 16, 1]. The mean (μ) and standard deviation (σ) were calculated for each set to assess the distribution of input data per channel such that each location had two arrays, each of size [1,16], for mean and standard deviation per channel. The potential for these metrics to predict model performance and the portability of a CCM is explored and discussed in later sections.

3. CNN architecture

The CNN used in this study was built, trained, and evaluated using PyTorch (v1.3.1) and a 16-core AMD CPU and 2 GB MSI GTX 1050 GPU. The goal was to develop a base model with fast training times and good performance on training and validation sets, rather than full optimization of CNN architecture. This led to some focus on regularization techniques to avoid overfitting, such that the model balances complexity with robust performance across locations of climactic and geographic diversity. The process of model tuning is outlined in detail in Appendix A. An overview of the final CNN structure is given in Table 2. The final model structure contains 160,510 trainable parameters. Though the number of parameters is relatively small for a deep learning network, results show that a CNN does not need to be very large to improve on a deterministic cloud mask.

4. Results and discussion

In the following sections, performance of the CCM is evaluated with respect to the asymmetry of model portability and the impact of increasing data diversity in training. Three categories of performance are discussed: (1) CCM trained and applied to the same location, (2) CCM trained on one location and applied to a new location, and (3) CCM trained on a combination of two locations and applied to a third, new location.

A brief note on notation, the subscripts i, j, k∈[1,12] for the 12 locations used in this study. The right arrow (→) is used to differentiate location(s) that a model was trained on versus the location to which it was applied. The term MCC_i→j indicates the average MCC score for a model trained on data from location i applied to itself (i→i). Likewise, MCC_i→k indicates the average MCC score for a CCM trained on a combination of two location’s training data i→j then applied to the test set of a new location k.

4.1. Baseline model

In the baseline model, the CCM is trained on, then applied to test data from the same location (i→i). This is possible if a ground station is available to create target data for training. ACC and MCC scores of the ACM and CCM are compared in Table 3. ACC scores are provided as
Table 3
Numerical values of the ACM and CCM accuracy (ACC) and MCC score for each location. A standard deviation from 10 iterations is given for each CCM score. The CCM shows improvement in every location in both ACC and MCC.

<table>
<thead>
<tr>
<th>Location</th>
<th>ACC ACM</th>
<th>ACC CCM</th>
<th>MCC ACM</th>
<th>MCC CCM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BON</td>
<td>0.699</td>
<td>0.890 ± 0.006</td>
<td>0.390</td>
<td>0.710 ± 0.023</td>
</tr>
<tr>
<td>DRA</td>
<td>0.794</td>
<td>0.876 ± 0.004</td>
<td>0.591</td>
<td>0.751 ± 0.007</td>
</tr>
<tr>
<td>FPK</td>
<td>0.839</td>
<td>0.893 ± 0.005</td>
<td>0.631</td>
<td>0.723 ± 0.015</td>
</tr>
<tr>
<td>GWN</td>
<td>0.753</td>
<td>0.899 ± 0.006</td>
<td>0.537</td>
<td>0.721 ± 0.009</td>
</tr>
<tr>
<td>PSU</td>
<td>0.829</td>
<td>0.901 ± 0.010</td>
<td>0.517</td>
<td>0.707 ± 0.014</td>
</tr>
<tr>
<td>SFE</td>
<td>0.855</td>
<td>0.904 ± 0.009</td>
<td>0.560</td>
<td>0.685 ± 0.051</td>
</tr>
<tr>
<td>TBL</td>
<td>0.834</td>
<td>0.910 ± 0.006</td>
<td>0.596</td>
<td>0.768 ± 0.013</td>
</tr>
<tr>
<td>ABQ</td>
<td>0.816</td>
<td>0.909 ± 0.006</td>
<td>0.655</td>
<td>0.800 ± 0.012</td>
</tr>
<tr>
<td>BIS</td>
<td>0.854</td>
<td>0.920 ± 0.006</td>
<td>0.515</td>
<td>0.666 ± 0.018</td>
</tr>
<tr>
<td>HNX</td>
<td>0.797</td>
<td>0.860 ± 0.008</td>
<td>0.636</td>
<td>0.735 ± 0.015</td>
</tr>
<tr>
<td>SLC</td>
<td>0.824</td>
<td>0.893 ± 0.006</td>
<td>0.661</td>
<td>0.742 ± 0.022</td>
</tr>
<tr>
<td>STE</td>
<td>0.818</td>
<td>0.904 ± 0.006</td>
<td>0.474</td>
<td>0.639 ± 0.027</td>
</tr>
<tr>
<td>AVG</td>
<td>0.809</td>
<td>0.897 ± 0.023</td>
<td>0.564</td>
<td>0.721 ± 0.058</td>
</tr>
</tbody>
</table>

4.2. Transfer learning

In many cases, ground station telemetry may not be available to create a ground truth data set for training. The next best option would be to use a CCM trained on another location where ground station telemetry is available. As discussed in Section 1.2.1, this strategy is known as transductive transfer learning. Before discussing transfer learning performance, the authors seek to introduce a framework used to characterize the combinations of locations involved in training and application. The framework is derived from baseline model performance results and discussed in the following section. The performance of the transfer learning models is then discussed first for models trained on single locations at \( i \rightarrow k \) (Section 4.2.2), then on paired locations as \( i/j \rightarrow k \) (Section 4.2.3).

Results show that transfer learning is asymmetric according to input characteristics of the respective source and target data sets and that greater diversity in training sample can improve performance.

4.2.1. Predicting performance

During analysis, it was observed that the CCM achieves higher performance in locations with relatively high percentage of clear samples. Fig. 2(a) demonstrates this relationship. MCC_\( i \rightarrow j \) is shown relative to clear sample percentage in the training set for a given location. Vertical bars represent the standard deviation from a set of 10 baseline models. Locations with a larger percentage of clear sky samples achieve higher performance through the CCM.

In a realistic transfer learning application, evaluation of the percent clearness of a target location relative to the source location may not be possible to ascertain through the same assessment of ground telemetry. Thus, a predictive metric derived solely from input data was preferred to the clearness metric.

It was found that mean and standard deviation per channel from training data samples captured similar characteristics as the percent clearness metric. The per channel sampling strategy is described in Section 2.4.

Fig. 3 shows the Pearson correlation coefficient (\( r \)) of MCC_\( i \rightarrow j \) with the mean and standard deviation of each channel, differentiated by color. The magnitude of the correlation (|\( r \)|) is given on the \( y \)‐axis with direct and inverse proportionality differentiated by hatch pattern. Notably, the two visible channels and the first near-IR channel (channels 1 through 3) show the largest correlation magnitudes. The largest positive correlation values in descending magnitude order are means of channels 1, 2, and 3. The largest negative correlation values in descending magnitude order are standard deviations of channels 2, 3, and 1.

Channel 1 mean and standard deviation were selected to characterize transfer learning performance. MCC_\( i \rightarrow j \) is plotted against channel 1 mean and standard deviation in Fig. 2(b) and (c). The correlation coefficient, \( r \), for each linear fit is given per subplot. Locations with higher mean channel 1 values and lower channel 1 variance achieve better performance. The fit of the linear regression for channel 1 means is similar to that of the fit using percent clearness. Similar result features could be shown with channels 2 and 3. Correlation coefficients beyond channel 3 are substantially lower but could potentially provide additional information that would improve the linear regression fit. For example, a sum of the means of channels 1, 2, and 3, weighted by their respective standard deviations as \( \sum_{i=1}^{3} \frac{|1| \cdot \mu_{i,a,b,r}}{|3|} \) results in a correlation coefficient of \( r = 0.745 \), a slight improvement on the correlation to \( \mu_{i,a,b} \), alone, which is \( r = 0.740 \). A full exploration of potential input metric combinations and appropriate weighting is beyond the scope of this study. In addition, further analysis could benefit from an increased number of locations or from varying location input characteristics through careful tuning of location input data combinations. For this analysis, using only channel 1 mean and standard deviation were found to be sufficient and are used to predict performance in the following transfer learning scenarios.
4.2.2. Single location training

Transfer learning is first evaluated on CCM models trained on data from only one location, termed single location training and denoted by $i \to k$ such that $i \neq k$. Average MCC scores from a set of 10 models are shown in Fig. 4, where the training, or source, locations are shown on the vertical axis and target locations are shown on the horizontal. Locations are ordered by increasing channel 1 means. The $MCC_{i \to i}$ scores, located on the diagonal, are included for comparison but are not, themselves, instances of transfer learning. In all but one case (HNX), $MCC_{i \to i}$ is always greater than $MCC_{k \to i}$. Scores tend to be lower on the upper right than corresponding scores across the diagonal.

That is to say, portability is asymmetric and, in general, $MCC_{i \to k} > MCC_{k \to i}$ when $\mu_{i, ch1} > \mu_{k, ch1}$.

This relationship is reinforced in Fig. 5, which shows performance of the transfer model relative to the difference between channel 1 source and target mean ($\mu_{i, ch1} - \mu_{k, ch1}$), shown on the x-axis, and standard deviation ($\sigma_{i, ch1} - \sigma_{k, ch1}$), denoted by the color scale. The first subplot shows $MCC_{i \to k}$ scores and the second subplot shows a penalty score, defined as

$$P_{i \to k} = \frac{MCC_{k \to i} - MCC_{i \to k}}{MCC_{k \to i}}.$$ (2)
Fig. 3. Correlation coefficients between MCC$_{i ightarrow i}$ and per channel mean and standard deviation. Channels 1 through 3 have the largest positive correlations between MCC$_{i ightarrow i}$ and mean and the largest negative correlations between MCC$_{i ightarrow i}$ and standard deviation.

Fig. 4. MCC scores of baseline model ($i ightarrow i$) and single location transfer learning ($i ightarrow k$).

Fig. 5. Showing asymmetry of transfer learning: (a) MCC scores from $i ightarrow k$ models and (b) penalty of $i ightarrow k$ models in comparison to $k ightarrow k$ models. The penalty score is defined in (2) and captures the effective loss in performance of single location transfer learning, when target data location is not available for training.

which should be interpreted as the loss in performance observed in using the CCM when target location data is not available during training. In a perfectly robust model, $P_{i ightarrow k} = 0$, indicating no loss in performance relative to a baseline model. While, realistically, models applied to unseen target location data have worse performance than models trained on the target location data, a number of transfer learning scenarios have
low penalty scores. Thus, for some cases, it is appropriate to use transfer learning with a single-location-trained CCM if target location data is unavailable.

The proposed metrics to characterize transfer learning scenarios are defined as follows:

\[ U_{i \rightarrow k, \text{ch}1} = \frac{\mu_{i, \text{ch}1} - \mu_{k, \text{ch}1}}{\mu_{k, \text{ch}1}} \]  

(3)

\[ V_{i \rightarrow k, \text{ch}1} = \frac{\sigma_{i, \text{ch}1} - \sigma_{k, \text{ch}1}}{\sigma_{k, \text{ch}1}} \]  

(4)

These metrics can be interpreted as normalized differences between the characteristic mean and variance of source and target channel 1 values, with respect to the target location.

Fig. 6 maps the transfer performance of the CCM as related to the difference in channel 1 average and variance between the training location and the applied location. Each marker in the scatter plot represents a set of 10 CCMs trained on location \( i \) and applied to location \( k \), where \( i \neq k \), such that there are 132 unique instances of \( i \rightarrow k \). Marker size and transparency in the scatter plot correspond to increasing standard deviation from the set of 10 CCMs. High performing CCMs are located in the center to lower right of the scatter plot.

![Fig. 6](image)

**Fig. 6.** Showing transfer performance as related to the normalized difference in channel 1 average \((U_{i \rightarrow k, \text{ch}1})\) and variance \((V_{i \rightarrow k, \text{ch}1})\) between source and target locations. Relatively high performing CCMs are located in the center to lower right of the scatter plot.

### 4.2.2. Combined location training

Given results from single location training, there is clear potential to achieve good model performance especially when metrics like channel 1 mean and standard deviation align with the appropriate direction of transfer learning, from high to low mean and from low to high variance. This result persists when training on a combination of training samples from two locations, termed combined location training as \( i \rightarrow k \) where \( i \neq j \neq k \). The available 12 locations in this study allows a total of 66 unique combinations of location pairs, with a remaining 10 locations available for testing as target locations.

To combine locations for this experiment, training sets were appended then reduced. A reduction was applied to ensure that combined location training data was not inherently advantaged over single location training by increased sample size. For a combination of two locations, every other sample in the combined training sets were selected.

To characterize the input metrics in combined location training cases, 10,000 samples were again selected from the combined location data, such that equal sampling was still performed on input data that the CCM would see. Mean and standard deviation were then calculated using the same process described in Section 2.4.

Fig. 8 follows the form of Fig. 6 but now maps the transfer performance of CCMs trained on a combination of two locations and applied to a third. Again, each marker in the scatter plot represents a set of 10 identically trained CCMs of the form \( ij \rightarrow k \), resulting in 660 data points. Higher scores are collected in the lower right of the scatter plot.

**Table 4**

<table>
<thead>
<tr>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U_{i \rightarrow j} )</td>
<td>( V_{i \rightarrow j} )</td>
<td>( V_{i \rightarrow j} )</td>
</tr>
<tr>
<td>(a)</td>
<td>(b)</td>
<td>(a)</td>
</tr>
<tr>
<td>0.299</td>
<td>0.003</td>
<td>0.427</td>
</tr>
<tr>
<td>0.279</td>
<td>0.001</td>
<td>0.391</td>
</tr>
<tr>
<td>-0.382</td>
<td>-0.382</td>
<td>0.617</td>
</tr>
<tr>
<td>-0.355</td>
<td>-0.355</td>
<td>0.572</td>
</tr>
<tr>
<td>-0.013</td>
<td>0.219</td>
<td>0.755</td>
</tr>
<tr>
<td>-0.004</td>
<td>0.154</td>
<td>0.758</td>
</tr>
</tbody>
</table>

4.2.2. Combined location training

On average, training on two locations improves CCM performance. The average MCC score of all 132 models in the single location training is 0.416. The average MCC score of in 660 combined location models is 0.494. If we reduce single location and combined location models to only those where transfer learning satisfies \( U_{i \rightarrow j} > 0 \) and \( V_{i \rightarrow j} < 0 \), (b) columns in Table 4, then the average scores for single and combined location training are 0.527 and 0.580, respectively. Both actions of (1) increasing the diversity in training data sets from one to two locations and (2) using \( U_{ij \rightarrow k, \text{ch}1} \) and \( V_{ij \rightarrow k, \text{ch}1} \) to assess favorable transfer conditions, result in better performance scores.

In the case of combined location transfer learning under favorable conditions, the average MCC score of 0.580 exceeds the average ACM MCC score of 0.564 on all locations. While this comparison has its limitations, given that target locations are not uniformly represented in the subset of favorable transfer learning scenarios, the average ACM score serves as a reference point to evaluate the suitability of predictive performance metrics \( U_{ij \rightarrow k, \text{ch}1} \) and \( V_{ij \rightarrow k, \text{ch}1} \) in identifying favorable conditions. The (a) columns in Table 4, where all combinations are represented, \( i \rightarrow k \) rows have 132 combinations and \( ij \rightarrow k \) rows have 660. The (b) columns, where sets are reduced to only favorable transfer learning directions as determined by the sign on \( U \) and \( V \), \( i \rightarrow k \) rows have 58 combinations and \( ij \rightarrow k \) rows have 263. By adding a location, the ranges of \( U \) and \( V \) shrink slightly from 0.726 to 0.670 in \( U \) and
5. Conclusion

This work evaluates the performance of a convolutional neural network (CNN)-based cloud mask (CCM) at 12 geographically and climatically diverse locations across the continental U.S. (CONUS). Performance is largely characterized by the Mathews correlation coefficient (MCC) score. The accuracy (ACC) score plus a set of true positive, false positive, false negative, and true negative rates (TPR, FPR, FNR, and TNR) are used to assess the baseline performance of the CCM in comparison to the Advanced Baseline Imager (ABI) cloud mask, here designated as ACM. The baseline CCM showed an average improvement of 11% in ACC and 30% in MCC and exceeded the ACC and MCC scores of the ACM in every location. More specifically, the baseline $i \rightarrow k$ CCM showed 2.5% improvement in clear sky identification rate (TPR) and 15.7% improvement in cloudy sky identification rate (TNR) across locations.

The potential for transfer learning is explored through single location training ($i \rightarrow k$) and combined location training ($ij \rightarrow k$). Results show that training on multiple locations improves the average transfer performance of the CCM. The total average MCC score of all transfer cases considered is 0.416 for single location training and 0.494 for combined location training. Adding a second location in training improved average performance by 18.8%

Results show that channel 1 average and standard deviation values provide good predictive value of baseline CCM performance. Channel 1 mean and standard deviation values are then used to construct a normalized difference in mean and variance between source and target locations for the purpose of building the transfer learning schemes. A methodology for predicting the success of transfer learning schemes is outlined through the introduction of two input metrics, $U_{ij \rightarrow k, ch1}$ and $V_{ij \rightarrow k, ch1}$. When viewing MCC scores against these metrics, the asymmetry of transfer learning becomes apparent as shown in Figs. 5, 6, 7, and 8. Models with higher average and lower variance, where $(ij) \rightarrow k \in \{ (U_{ij \rightarrow k} > 0, V_{ij \rightarrow k} < 0) \}$, show greater average performance than their counterparts. In cases where these favorable conditions are met, the total average MCC score is 0.527 for $i \rightarrow k$ and 0.580 for $ij \rightarrow k$. The 0.580 average of MCC $ij \rightarrow k$, from a set of 263 combinations of favorable transfer conditions, is greater than the average ACM MCC score, which is 0.564 for the 12 locations.

This study shows that CNN-based cloud detection methodologies offer great potential for improvement of deterministic cloud mask schemes such as those currently used in standard remote sensing products. The portability of the CCM using only a few calibrating locations with high range of input values also offers the possibility of developing CCMs that can be used for target regions in widely different climates.
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 gratefully acknowledge partial support from the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE), USA under Solar Energy Technologies Office (SETO), USA Agreement No. EE0008216.

Appendix A. CNN architecture tuning

As is common with hyperparameter tuning (Smith, 2018), a process of trial and error was used to reach a collection of parameters with good performance across different locations. Tuning involved multiple iterations of training the model on training set samples then evaluating the trained models on the validation set. No samples from the test sets were used to determine model parameters. A scheme was configured to test the effect of varying a single parameter with multiple iterations under the same suite of hyperparameters, then adjusting one parameter at a time to understand its significance. Because many of these hyperparameters are strongly interdependent (Smith, 2018), this linear approach to tuning was used on a limited number of parameters and served as secondary validation in parallel to the broader trial and error tuning process. Model tuning was carried out using DRA, PSU, and BON data to capture both the clearest (DRA) and cloudiest (PSU) training sets, plus one intermediate training set located between DRA and PSU in terms of both percent clearness and geographic position. Hyperparameter tuning using these three locations was sufficient to produce similar outcomes in the other 9 locations.

The basis of this CNN is similar to the CNN structure outlined in Sun et al. (2018), which contains two convolutional-pooling (CP) layers and two fully-connected (FC) layers. The number of CP and FC layers were explored through the tuning process; however increasing to greater than two layers for either section was not shown to have significant improvement in performance. A mini-batch of 256 was used for all locations and all models trained for 100 epochs. Cross entropy loss and the rectified linear unit (ReLU) were used as the loss and activation functions, respectively. To mitigate class imbalance, weights were assigned in the loss function proportional to inverse class frequency in the training set. The AdamW algorithm (Loshchilov and Hutter, 2019) was used for optimization with a weight decay of 0.01, consistent with the algorithm’s default value.

A cyclic learning rate scheduler was employed to improve robustness in training (Smith, 2017). In general, small (less than $10^{-\text{2}}$) learning rates showed best performance on the validation sets. The base and maximum learning rates used are $10^{-\text{5}}$ and $10^{-\text{4}}$, respectively. A step size of 500 was used and the cycle momentum feature was not employed.

A layer of 1-D batch normalization (Ioffe and Szegedy, 2015) and drop out (Hinton et al., 2012) were added prior to each fully-connected layer to improve regularization and reduce potential for overfitting. The combination of these two layers in sequence is termed the Independent-Component (IC) layer (Chen et al., 2019). Performance improvement was found when IC layers were placed before the fully-connected layers in the CNN.

After converging on structure and values regarding the preceding components, effects of first convolutional layer width are explored. The first convolutional layer depth determines all convolutional layer complexity as the number of layers is doubled in each subsequent layer. Fig. A.9 shows the range of validation set scores after training on a two-layer convolutional-pooling CNN with 1 to 36 filters in the first layer. In this tuning experiment, all other parameters were held constant.

![Fig. A.9. Parameter search space for the network width of the first convolutional layer. Each box and whiskers represent 10 iterations of a CCM at the given network depth.]

Because of stochasticity inherent to CNN training, CNN models used for hyperparameter tuning and performance evaluation were trained and evaluated in sets of 10, hence the boxplots in Fig. A.9 that show variation in the 10 model iterations. The center line represents the median while the box indicates the interquartile range and the whiskers extend to 1.5 times the interquartile range. Ideally, this variance should be small to ensure reliability of performance.

In addition to hyperparameter settings mentioned here, the final CNN model architecture is given by Table 2.

Appendix B. Discussion of binary metrics

Classification performance metrics facilitate the comparison of different models across data sets and training parameters. Interpreting the relative success of classification schemes may be strongly influenced by choice of classification metric. Some metrics are exclusively defined on correct identification of the positive class and therefore overestimate performance of models that are extremely good at identifying positive targets at the expense of negative ones (Brown, 2018). However, in particular applications, this negative class invariance may be appropriate for performance evaluation (Sokolova and Lapalme, 2019). Ultimately, evaluation metrics should be chosen to suit the priorities and performance requirements of the specific application.

In binary classification, the target values can be defined as either positive (P) or negative (N). A correctly identified positive sample is a true positive (TP). An incorrectly identified positive sample, where the model incorrectly classified a negative sample as positive, is defined as a false positive (FP). The same logic follows for true negative (TN) and false negative (FN).

True positivity rate (TPR) is the number of correctly identified positive classes, and is also known as recall or sensitivity. TPR is defined as

$$TPR = \frac{TP}{TP + FN}. \quad (B.1)$$

The corresponding metric for the negative class is the true negativity rate (TNR), defined as

$$TNR = \frac{TN}{TN + FP}. \quad (B.2)$$

Each of these has their corresponding negative counterpart: false positivity rate (FPR) and false negativity rate (FNR). These are defined as

$$FPR = 1 - TNR = \frac{FP}{TN + FP}. \quad (B.3)$$

$$FNR = 1 - TPR = \frac{FN}{TP + FN}. \quad (B.4)$$
Fig. B.10. Comparison of ACC, F1, and MCC score distribution on a range of TPR and TNR under two distinct scenarios: (1) top row, a balanced two-class data set and (2) bottom row, an unbalanced two-class data set.

\[
FN_R = 1 - TPR = \frac{FN}{TP + FN}.
\]  

(B.4)

Positive predictive value (PPV), also called precision, is defined as

\[
PPV = \frac{TP}{TP + FP}.
\]  

(B.5)

PPV combined with TPR forms the F1 metric, or F-measure, a commonly used standard in machine learning (Chicco and Jurman, 2020). F1 is defined as

\[
F1 = \frac{2 \times PPV \times TPR}{PPV + TPR} = \frac{2 \times TP}{TP + (2 \times TP) + FN}.
\]  

(B.6)

and represents the harmonic mean of precision and recall. It is usually seen as a less-biased metric than accuracy when there is class imbalance (Zhang et al., 2015b). F1 biases evaluation on the positive class as its numerator is only evaluated on true positive prediction metrics. To compensate, some studies recommend combined macro- and micro-averaging (Zhang et al., 2015b) or class swapping in addition to macro- and micro-averaging (Chicco and Jurman, 2020). Though these practices improve weaknesses in the F1 score, they are infrequently adopted (Chicco and Jurman, 2020) and F1 continues to be a commonly used metric when evaluating binary classification.

Accuracy (ACC), sometimes termed event prediction accuracy (EPA) or event accuracy (EA), is the total ratio of correctly predicted samples to the total number of samples, defined as

\[
ACC = \frac{TP + TN}{TP + TN + FP + FN}.
\]  

(B.7)

Accuracy is one of the most commonly used metrics notable for its invariance to performance on a particular class (Sokolova and Lapalme, 2019). The ACM technical document defines accuracy as a probability of detection (POD), or total correct decisions over total decisions (Heidinger and Straka III, 2013). Other studies report specific class accuracy metrics as in POD for clouds and POD for clear skies (Jimenez, 2020), akin to providing components for macro-averaging.

The Matthews correlation coefficient (MCC) is widely suggested for evaluation of binary classification with unbalanced classes (Chicco and Jurman, 2020; Brown, 2018; Sokolova and Lapalme, 2019; Luque et al., 2019). The metric has a range of 1 to 1 and is defined as

\[
MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}.
\]  

(B.8)

An MCC of 0 suggests complete randomness, while a value of 1 indicates perfect positive correlation and 1 indicates perfect inverse correlation (Chicco and Jurman, 2020). To demonstrate the impact of class imbalance, we present a simple example. Imagine a data set of 10,000 samples, of which 1,000 are clear and 9,000 are cloudy. If a model arbitrarily returns a cloudy classification, regardless of input, the model would achieve 90% accuracy. Its specific confusion matrix metrics would be: \(TP = 0\), \(TN = 9,000\), \(FP = 0\), \(FN = 1,000\), which leads to an MCC score of 0.

The relative balance of clear and cloudy conditions varies greatly from location to location and is often skewed toward cloudy. Only the most desert-like locations show percent clearness near half of the time, which approximates balanced classes. Because the classes are unbalanced, the metrics that evaluate on true positives tend to skew scores toward performance on positive identification. In this case, correct identification of both the clear and cloudy images is considered important. There is no preferred identification nor default assumption. Arbitrarily, in this CNN, the clear conditions are labeled as positive, meaning a true positivity rate would be defined by the number of correctly identified clear samples. For a data set with very few positive samples available, metrics that skew toward performance on positive classes will over-penalize a model that performs poorly on positive identification, even if the rate of correct negative (or cloudy) identification is high.

Fig. B.10 is adapted from Brown (2018) to visualize the performance metric spread on unbalanced classes. In the balanced case (top row of Fig. B.10), score distributions look similar. Only models that have high TPR and TNR reach top scores based on the ACC, F1, and MCC metrics. The differences are more apparent in the unbalanced two-class case (bottom row). In ACC and F1, models can achieve high scores if performance is strong in the dominant class alone, regardless of the minority class performance. The score distributions emphasize the robustness of the MCC score in unbalanced data sets.


