Cloud cover estimation and prediction from sky images

M Aishwarya
ai21resch11002@iith.ac.in

December 3, 2021

Abstract

To achieve the goal of making solar energy a reliable source of energy it is necessary to estimate and predict cloud cover in advance. This is challenging due to changes in lighting, pollution levels, weather, etc. Using total sky images we try to first estimate and then predict cloud cover in the near future (10-40 min ahead). Image segmentation with U-Nets showed promising results with the ability to adapt to various sky conditions as long as the training data is diverse and representative. We show that Convolutional Long Short term memory models can be used to predict future instances of sky images and use that to predict cloud cover.

1 Introduction

One of the main challenges for wide scale adoption of solar energy is its unreliability due to changes in local weather conditions. The presence of clouds, their position and movement has a big impact on the amount of solar energy that can be captured. It is necessary to be able to predict beforehand when the energy output is going to be low, so as to prepare and optimize usage of available power, until the energy output is more consistent. Hence this project aims to develop models capable of estimating percentage of cloud cover from sky images at present and at future intervals like 10 to 40 minutes ahead so as to provide alerts in advance. The energy output from the sun is measured in terms of solar irradiance with units Watts/meter^2 (energy per unit area). Clouds play an important role in determining the amount of solar irradiance available. Cameras facing the sky (Total Sky imagers) can be used to capture the sky images in regular intervals. If we can use these images to estimate cloud cover at present and use that to predict how the cloud cover will change in the next few hours, it will serve as the first step in the process of predicting solar irradiance and will help optimize use of energy and increase acceptance and transition to solar energy as we can estimate its reliability in advance. This project idea is inspired by the Shell.ai Hackathon

2 Problem Statement

To develop a model to estimate percentage of cloud cover in a given image of the sky. Further by analysing how the cloud cover changes in the past few minutes, to develop a predictive model to estimate cloud cover after few minutes. The outcome will be to get a prediction of future cloud cover which will help in taking appropriate measures to ensure constant supply of energy. The evaluation metric we used for cloud cover estimation is sparse categorical crossentropy, mean absolute percentage error and for prediction we use binary cross entropy error.

3 Literature Review

Described next are few of the approaches taken to address this problem of estimating and predicting cloud cover and movement. [2] Here future cloud coverage is predicted as an image by training deep convolution neural network models. It is approached in two

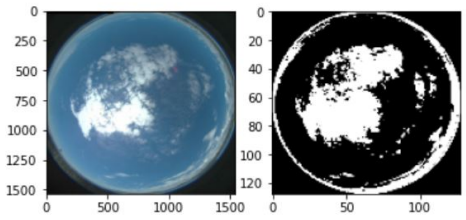
ways, first 4 sequential images are used to predict the next images and another approach which uses the 8 sequential images to predict the next 8 images in future. Generative Adversarial networks are used where the generator is fed 4 sequential real sky images and is tasked with generating the 5^{th} realistic fake image and the discriminator is fed these 4 sequential images along with the 5^{th} images that can either be the real images or the output of the generator. The discriminator then classifies the input as real or fake(1/0). The goal is to teach the generator to produce realistic sky images which can be used to estimate future cloud cover. The criteria used to classify whether a pixel is part of the cloud or sky was developed specific to the dataset used by a specialist. The average mean square error was found to be less than 5%. This approach is interesting as it is solely dependent on the sequence of input images, Here the images were sample with an interval of a minute, that is images of last 4 minute the cloud cover at the 5^{th} minute is predicted. We can try experimenting with this approach by increasing the interval between sequential images to try predicting further into the future. An iterative approach was also tried where the generated image along with the last 3 images were use to predict the next state. This was done for 8 sequential images, thus predicting up to 8 minutes later. In the second approach a sequence of 8 images were fed to the generator to predict the next 8 images. This was a time-consuming approach but gave good results till 7-10 frames ahead, depending on the contribution of adversarial loss. [4] here they used motion tracking concept to predict movement of clouds from sky images. Sky images were pre-processed by taking the difference of blue and red channels then Shi-Tomasi algorithm was used to detect points that form edges and corners of clouds. Then these points used as input to Lucas-Kanade optical flow algorithm, where the points are tracked over 20 sequential images. The output was a set of (x,y) coordinates for each point tracked. These points were then used to train an artificial neural network (ANN) in the following manner. First the 20 frames were divided as 10 frames for training and 10 frames for testing. Then the data was split into features and target by taking the coordinates for a point in the first four frames as input and the coordinate in the 5^{th} frame as the target. This process is repeated for all points of interest identified in the image to generate the dataset to train the ANN. The mean absolute percentage error was around 6 and 23 for x and y coordinates respectively. It was observed that there was more uncertainty in the prediction of y coordinates as the input images was rectangular of dimension 600x900. The following papers focus on prediction of solar irradiance from sky images. This is actually the next step after cloud cover prediction by the approaches described can be used for our current problem statement. [5]Here the goal was to predict solar irradiance fluctuation which is used to alert changes in energy output in solar farms about 1-2 minutes in advance. This is relevant in our current scenario to predict cloud cover as the end goal is the same. Sky images were preprocessed by taking the sum of absolute difference between 3 channels, this was done to better differentiate the features in the images like sky and clouds. Then Hough transform technique used to draw a circle around the sun. Direction of movement of clouds were identified using the Farneback method. Final images of size 256x256 were created which consisted of the sun and vectors representing direction of movement of clouds. ResNet-101 a type of convolution neural network (CNN) was trained on these vector images to raise a trigger when the cloud vectors were moving in the direction of the sun. The method used to label the images into the two classes to train the model was not mentioned. Overall the model was able to correctly trigger alerts around 73% of the time. This paper helps us understand how we can use the movement of clouds as a feature for understanding cloud cover. The drawback is that images from different cameras/angles may be needed to track the sun as it moves throughout the day. [6]This paper proposes a combination of classification and regression models to predict the solar irradiance from sky images. Image preprocessing consisted of removing non sky regions/background from image. Cooper formula for solar radiation used to de-trend the data by dividing the irradiance by cos(solar zenith angle). This was done to eliminate the effect of sun position (time) in clear sky images. Images were manually labelled by providing both classification (0 = no shade,

1 = otherwise) and regression labels (solar irradiance values). A two-step approach was proposed where an Alex net model trained on images for classification was used as a transfer learning model for regression, by freezing the convolution weights and updating the fully connected layers to allow for more continuous values that better represent the image. The classifier obtained around 93% test accuracy and regressor model had root mean square error of around $130W/m^2$ which was lower than a direct regression model(without transfer learning). [9]A CNN (TSI + GHI) model is proposed for global horizontal irradiance prediction (GHI) that consists of two parts. First the CNN model which takes in an images and predicts the corresponding GHI value, this output combined with the GHI value of the past n minutes is used to predict the future GHI after n minutes. RMSprop optimizer used to train the model and loss function is mean square error. Dataset consists of data collected for 10 days. The RMSE for this model was lower than when only the CNN was used by providing an input of n images to predict the GHI n minutes later. This paper provided forecast 5 to 20 minutes ahead which is an improvement over other approaches which predicted only few minutes ahead. But as the dataset considered was small, more validation is needed.[1] This paper proposes a new hybrid method based on machine learning (ML) algorithms and daily classification technique to forecast 1 h ahead of global solar radiation in the city of Evora. Firstly, several comparative studies have been done between random forest (RF), gradient boosting (GB), support vector machines (SVM), and artificial neural network (ANN). These comparisons were made using annual, seasonal, and daily testing sets in order to determine the best ML algorithm under different meteorological conditions.The evaluation of the proposed ML algorithms was carried out using the normalized root mean square error (nRMSE) and the normalized absolute mean error (nMAE). The results of the seasonal comparison show that the RF model performs well for spring and autumn seasons with nRMSE equaling 22.53% and 23.42%, respectively. While the SVR model gives good results for winter and summer seasons with nRMSE equaling 24.31% and 8.41%, respectively. In addition, the daily comparison demonstrates that the RF model performs well for cloudy days with nRMSE=41.40%, while the SVR model yields good results for sunny days with nRMSE=8.88%. The results show that the daily classification technique enhances the forecasting accuracy of ML models.Also for short-term solar irradiance forecasting [8]This work aims at bringing innovative insights via a novel approach to irradiance forecasting using the Deep Learning framework, which constitutes an effective environment for a richer modelling of the cloud cover and its dynamics. The model performance has been evaluated using the Mean Square Error (MSE) as error metric and a persistence of the clear-sky index as a reference to calculate the corresponding skill score.The current architecture of the network presented in the paper could be further upgraded with recurrent units to facilitate the extraction of relevant features from the temporal aspect of the data.

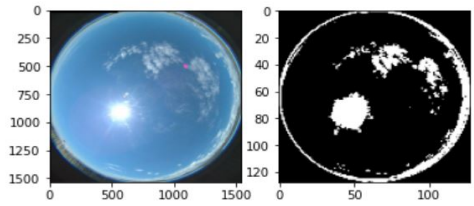
4 Preliminary Results

Described below are the various approaches that were tried to estimate and predict future cloud cover from sky images. First we look at methods to estimate current cloud cover from sky images. Then we try predicting 10 minutes ahead using a Long Short term memory model 4

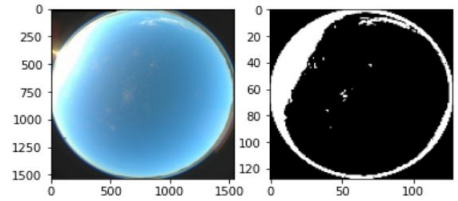
1. Pre-processing The sky image consist of a circular region with the image of the sky. We first apply a circular mask to image to retrieve only the pixels corresponding to the sky and make the rest black. Then we took the ratio of the red and blue channels of the image and set the pixel values to 1 when the ratio exceeds a predefined threshold and 0 otherwise. The threshold used here was 0.6, this number was obtained by experimenting on images with different thresholds, generally higher thresholds captured only very dense clouds and ignored other cloud covered regions. Then the image was resized to form a 128x128 image, this was done to make computation faster. Shown below are some example of the images before and after pre-processing



(a) Clouds



(b) Sun present



(c) Sun near edge

Figure 1: Comparing original and processed images

To calculate the percentage of cloud cover we use a function that computes the percentage of white and black pixels in the processed image .

Some observations with this method of pre-processing:

- The presence of sun and cloud was not clearly differentiated by this approach. That is, the sun can be misinterpreted as cloud in the processed image 1b
- As seen in the figure 1c presence of sun near the edge of the lens produces a distinctive image which can be misinterpreted as cloud.
- Also the depth of clouds were not differentiated. That is dense or sparse/thin clouds were represented the same way 1a.

This makes it difficult to accurately estimate cloud cover with this approach. Next we try using a convolution neural network to learn the features from the images that will help in estimation.

2. Convolution Neural Network (CNN) A subset of the data set consisting of 500 images was used to experiment with this approach, Using the entire data set with over 3000 images took longer duration to train. The idea was to pass these images through a CNN and train a model to predict cloud cover percentage. The model structure is represented in figure 2.

Layer (type)	Output Shape	Param #
rescaling_1 (Rescaling)	(None, 128, 128, 3)	0
conv2d (Conv2D)	(None, 128, 128, 32)	896
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_1 (Conv2D)	(None, 64, 64, 64)	18496
max_pooling2d_1 (MaxPooling2	(None, 32, 32, 64)	0
conv2d_2 (Conv2D)	(None, 32, 32, 128)	73856
max_pooling2d_2 (MaxPooling2	(None, 16, 16, 128)	0
conv2d_3 (Conv2D)	(None, 16, 16, 256)	295168
max_pooling2d_3 (MaxPooling2	(None, 8, 8, 256)	0
conv2d_4 (Conv2D)	(None, 8, 8, 256)	590080
max_pooling2d_4 (MaxPooling2	(None, 4, 4, 256)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 512)	2097664
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1)	513

Total params: 3,076,673
Trainable params: 3,076,673
Non-trainable params: 0

Figure 2: Model architecture used for estimating cloud cover

The graph of training loss is shown in 3 where its observed that on training with 500 images (resized to 50x50 to make computation faster) for 200 epochs, the training accuracy was around 60%

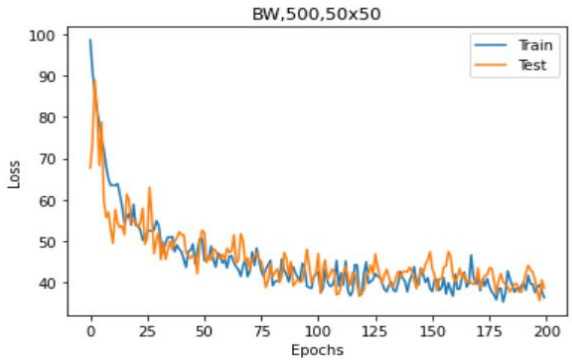


Figure 3: Model loss over epochs using processed grayscale images

Then the same procedure was followed for the original unprocessed RGB images, resized to 50x50, the training loss observed for each epoch is depicted in 4. This was done to observe if any loss of data due to pre processing affected the learning process. The training accuracy was around 70% but not reliable as the change in loss between epochs was varying drastically as observed in the figure4

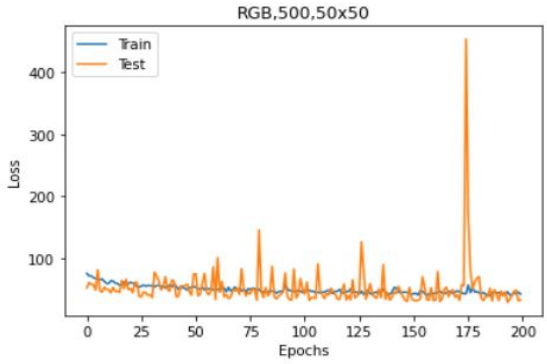


Figure 4: Model loss over epochs using RGB images

We can try increasing accuracy by experimenting with different model architectures and training for longer duration.

3. Convolution and Long Short term Memory (LSTM) Inspired by the idea described in this paper[2] where a generative adversarial network was developed to predict future sky images, we used a convolution LSTM model to predict the sky image 10 minutes ahead We used a sequence of 3 images (taken 10 minutes apart) to predict the 4th image in the sequence and use the function described previously **??** to count the percentage of white pixels and estimate cloud cover.The results on training the model on the sequence images of one month (22 sequences) are shown in fig 5.

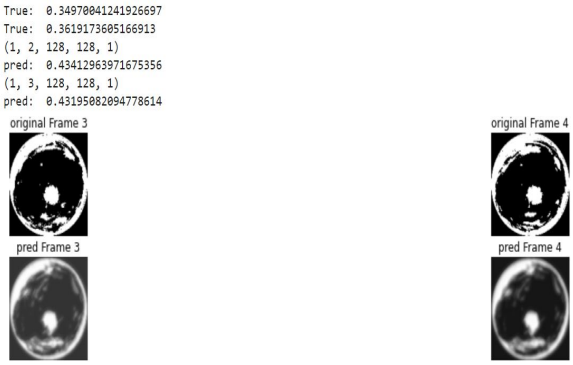
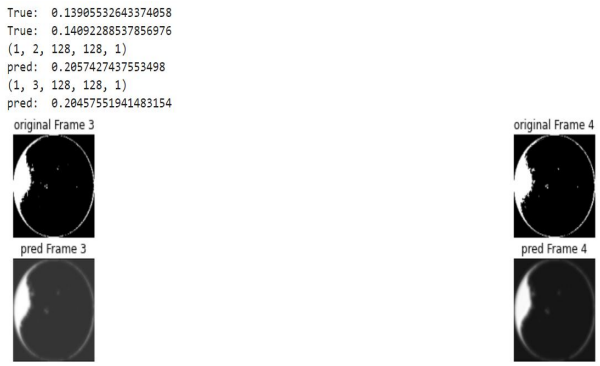
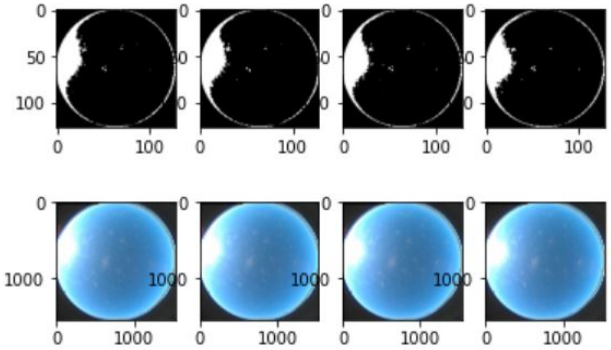


Figure 5: Comparing predicted and actual image



(a) Comparing predicted and actual image



(b) Actual sequence

Figure 6: Another example of data and predictions of model

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, None, 128, 128, 1 0
_________________________________________________________________
conv_lst_m2d (ConvLSTM2D)    (None, None, 128, 128, 64 416256
_________________________________________________________________
batch_normalization (BatchNo (None, None, 128, 128, 64 256
_________________________________________________________________
conv_lst_m2d_1 (ConvLSTM2D)  (None, None, 128, 128, 64 295168
_________________________________________________________________
batch_normalization_1 (Batch (None, None, 128, 128, 64 256
_________________________________________________________________
conv3d (Conv3D)              (None, None, 128, 128, 1) 1729
=================================================================
Total params: 713,665
Trainable params: 713,409
Non-trainable params: 256
_________________________________________________________________
```

Figure 7: Convolution LSTM model architecture

5 Proposed Approach

About the dataset: The dataset consists of sky images from a device called the total sky imager which consists of a fish eye lens pointed towards the sky that captures images at regular intervals. Images from near sunrise to near sunset were captured for 1 year at 10 minute intervals. Target variable which is percentage cloud cover was given for every minute throughout the year. This was sampled at 10 minute intervals and matched to the images. The method used to estimate the cloud cover and details on the devise used to capture the images, camera and lens specifications were not provided along with the dataset as it was part of a competition. Hence it was necessary to first determine a method to estimate cloud cover from an image and then after predicting the future instance of cloud configuration, use it to estimate future cloud cover. Also in the labels provided -1 was used to indicate night or late evening.

1. Processing
   - classify given sky image as cloudy, clear, night

     The reason for this step is to simplify the variety of data encountered for the prediction task. We need not predict future cloud cover states for late evening or night images as the sun sets. We can also avoid predicting when cloud cover is less than 10% as it may not drastically affect the amount of sunlight received.
2. Estimation
   - CNN regression

     Here we try estimating the cloud cover directly from the image without any thresholding. This is an attempt to try how CNN fit to this task. The preprocessing steps taken were: Applying a circular mask to remove boundary landscape and distortion due to lens. Then images were resized to 128x128 and normalized. The loss function used

for training was mean absolute percentage error

- apply segmentation to estimate percentage cloud cover

  Building on the thresholding approach, one of the major drawback was the selection of threshold, which cannot be uniform throughout the year. So we use a segmentation approach to train a model to classify a given pixel as cloud or not and count the total number of cloud pixels to estimate the cloud cover. This was inspired by the approach take by [10]

3. Prediction

   - Use Convolutional LSTM to predict future frames

     The task of predicting future cloud cover is challenging, few approaches use motion estimation [3] and optical flow algorithms [4], [5] which work best for few minutes up to 30 minutes ahead. [2] uses GANs which was trained to predict minutes ahead. In this project we use convolutional LSTM model to try to predict future frames from 10 to 40 minutes ahead.

6 Results and Discussion

1. classify given sky image as cloudy, clear, night

   To simplify the task of predicting cloud cover we need not consider night or late evening images as by then the sun sets. Also as the goal for prediction is to ensure no sudden/high impact disruptions are encountered, so images with less than 10% cloud cover are categorised as clear sky and not used. Remaining images are categorised as cloudy and are used for predicting future cloud cover. Some examples of the images that belong to each class are shown in 8

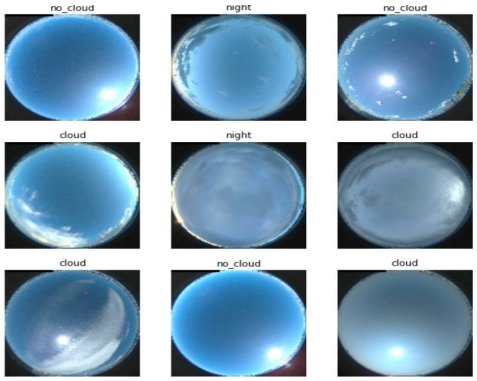


Figure 8: Examples of 3 classes

For this task of classification a Convolution neural network was used with 3 convolution and max pool layers followed by 2 dense layers 10. Model architecture inspired by [7]. The validation accuracy obtained was around 90% and training accuracy around 92%. The model was trained for 10 epochs. The loss and accuracy across epochs shown in 9 and the precision and recall for the three classes shown in figure 11. The label 'unk rain' is same as label night.

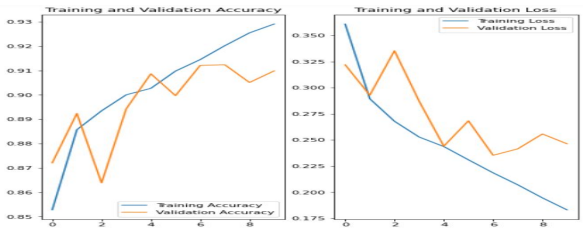


Figure 9: Loss and accuracy across epochs

Layer (type)	Output Shape	Param #
rescaling_2 (Rescaling)	(None, 100, 100, 3)	0
conv2d_6 (Conv2D)	(None, 100, 100, 16)	448
max_pooling2d_6 (MaxPooling2	(None, 50, 50, 16)	0
conv2d_7 (Conv2D)	(None, 50, 50, 32)	4640
max_pooling2d_7 (MaxPooling2	(None, 25, 25, 32)	0
conv2d_8 (Conv2D)	(None, 25, 25, 64)	18496
max_pooling2d_8 (MaxPooling2	(None, 12, 12, 64)	0
flatten_2 (Flatten)	(None, 9216)	0
dense_4 (Dense)	(None, 128)	1179776
dense_5 (Dense)	(None, 3)	387

Total params: 1,203,747
Trainable params: 1,203,747
Non-trainable params: 0

Figure 10: CNN model architecture

	precision	recall	f1-score	support
cloud	0.93	0.97	0.95	5879
no_cloud	0.85	0.66	0.74	1204
unk_rain	0.80	0.76	0.78	481
accuracy			0.91	7564
macro avg	0.86	0.80	0.82	7564
weighted avg	0.91	0.91	0.91	7564

Figure 11: Classification report

It is important to note that the percentage cloud cover provided as part of the dataset is not entirely accurate. For example the images shown in figure 12 were labelled as having more than 10% cloud cover whereas by human inspection no clouds or less that 10% cloud cover is observed

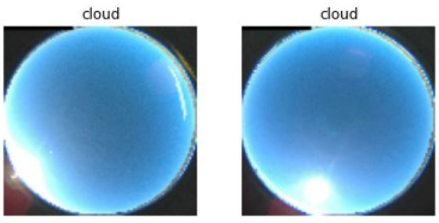


Figure 12: incorrect target labels

2. Convolution Regression for cloud cover estimation

This was the model that was trained to estimate the cloud cover directly from images. Figure 13 shows the changes in loss as the model trained over 20 epochs.

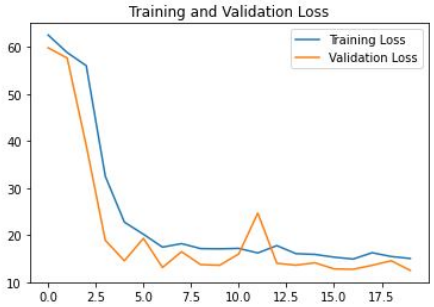


Figure 13: Train and validation loss over epochs

After 10 epochs, mean absolute percentage error loss was 17.0671 and validation loss: 13.5817. This values did not change much when trained for 20 epochs. The key difference in this approach compared to our preliminary approach was introducing changes in training dataset. Here only images that belong to the cloudy category (according to the first model) were used to train the model. This may have helped by making the dataset more consistent, which helped the model fit better. To reduce long training times only 50% of the dataset was used to train the model.

3. apply segmentation to estimate percentage cloud cover Inspired by [10] we tried a segmentation approach to estimate cloud cover. Since no pixel-wise annotation was available for the dataset at hand, we built and trained an encoder-decoder model to segment an image into 3 classes using the WSISEG database. Example of image and its segmentation in the database is shown in figure 14

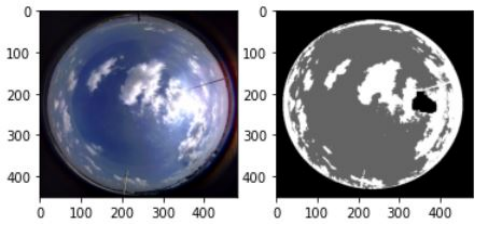


Figure 14: WSISEG data example of image and segmentation

Steps taken in preprocessing the images before training were: To resize the image to size 128x128, change segmented class values to 0,1,2 from 0,100,255 for simpler computation.

To construct the model the following architecture was used 15

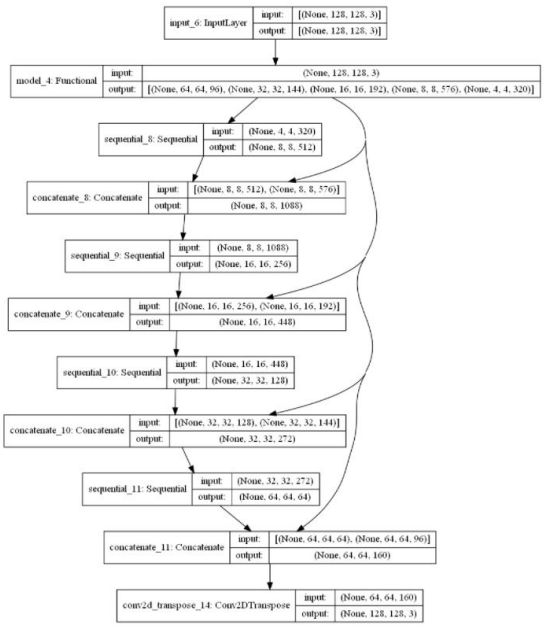


Figure 15: Segmentation model

Figure 16 shows the change in loss as the model was trained over 40 epochs

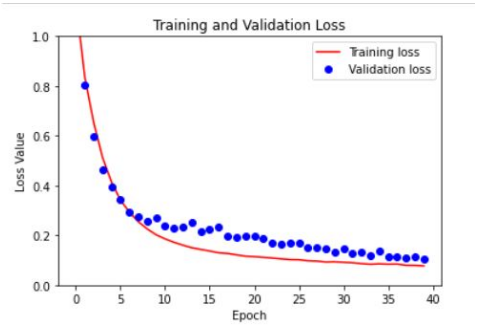


Figure 16: Train and validation loss over epochs

The validation accuracy obtained was 96.7%

Then we used this model to predict on our dataset and compared the results.

Examples of how the model performs segmentation on new dataset:

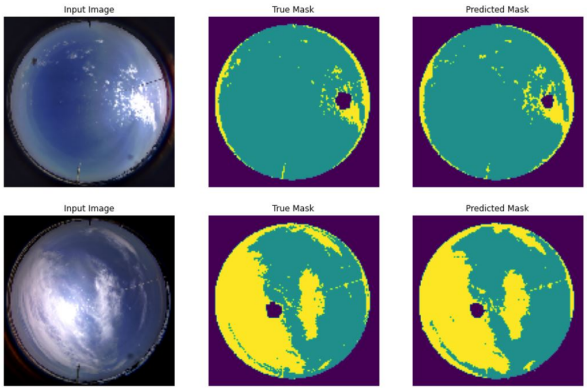
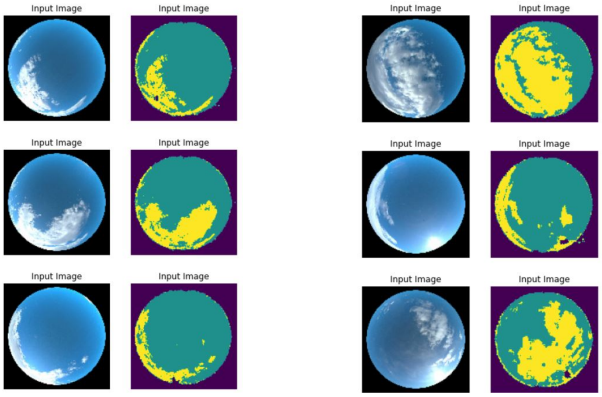


Figure 17: Segmentation of test image

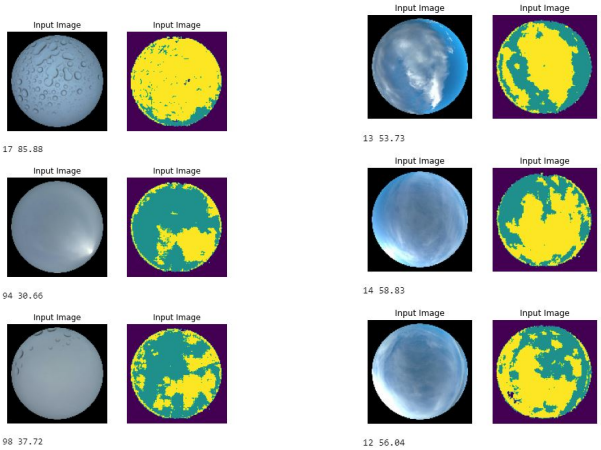
4. Convolutional LSTM to predict future cloud cover This approach was identical to our preliminary approach with the modification in the training dataset. Now only sequence of images that were classified as cloudy were considered and the data was set up as follows, the segmented version of sky images was used as the training data: Taking 8 sequential images the first 4 were considered as the training data and the next 4 were considered as the target data. This is equivalent



(a) difference in cloud cover between label and calculated below 5%

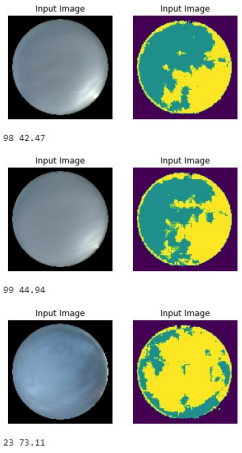
(b) difference in cloud cover between label and calculated around 20-30%

Figure 18: Examples correctly segmented



(a) rain

(b) thin clouds



(c) full clouds

Figure 19: Examples poorly segmented

to taking 4 sequential images taken 10 minutes apart (40 minutes) and use it to predict the next 40 minutes, in 10 minute intervals. To reduce training time only 10% of the data was used. The model was trained for 5 epochs and an improvement in predicted frames was observed from the second epoch to the fifth epoch as shown in the figures 20,21.

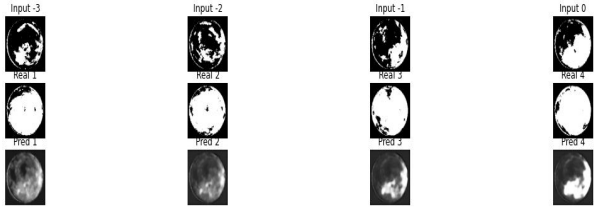


Figure 20: Model predictions after 2 epochs

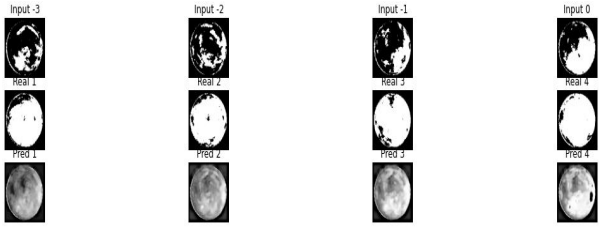


Figure 21: Model predictions after 5 epochs

The model predictions is not entirely accurate and more epochs may be needed to make it useful. The training was run for only 5 epochs due to computational limitations. It is interesting to note that although the input images were black and white, the predicted images are in gray scale. This may improve with more epochs or may need a threshold to convert it to black and white to compare with the target images. The binary cross entropy loss obtained for this model at the end was as follows: training loss was around 0.31, validation loss was around 0.3 and the learning rate used was 0.001.

Through this project we have come up with an approach to estimate and try to predict cloud cover from sky images. The prediction accuracy is expected to improve with more training time. To further improve accuracy for prediction other features like weather, solar elevation,etc can be considered along with images.

7 References

References

[1] Hamza Ali-Ou-Salah, Benyounes Oukarfi, Khalid Bahani, and Mohammed Moujabbir. A New Hybrid Model for Hourly Solar Radiation Forecasting Using Daily Classification Technique and Machine Learning Algorithms. In *Mathematical Problems in Engineering*, volume 2021, pages 1–12, 2021. 3

[2] George Andrianakos, Dimitrios Tsourounis, Spiros Oikonomou, Dimitris Kastaniotis, George Economou, and Andreas Kazantzidis. Sky image forecasting with generative adversarial networks for cloud coverage prediction. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–7, 2019. 1, 5, 7

[3] Chi Wai Chow, Bryan Urquhart, Matthew Lave, Anthony Dominguez, Jan Kleissl, Janet Shields, and Byron Washom. Intra-hour forecasting with a total sky imager at the uc san diego solar energy testbed. *Solar Energy*, 85(11):2881–2893, 2011. 7

[4] Ardan Hüuseyin Eşlik, Emre Akarslan, and Fatih Onur Hocaoğlu. Cloud motion estimation with ann for solar radiation forecasting. In *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, pages 1–5, 2021. 2, 7

[5] Thanonchai Leelaruji and Nithiphat Teerakawanich. Short term prediction of solar irradiance fluctuation using image processing with resnet. In *2020 8th International Electrical Engineering Congress (iEECON)*, pages 1–4, 2020. 2, 7

[6] Yanbin Lin, Dongliang Duan, Xuemin Hong, Xiaoguang Han, Xiang Cheng, Liuqing Yang, and Shuguang Cui. Transfer learning on the feature extractions of sky images for solar power production. In *2019 IEEE Power Energy Society General Meeting (PESGM)*, pages 1–5, 2019. 2

[7] Ryo Onishi and Daisuke Sugiyama. Deep Convolutional Neural Network for Cloud Coverage Estimation from Snapshot Camera Images. *SOLA*, 13(0):235–239, 2017. 7

[8] Quentin Paletta and Joan Lasenby. Convolutional neural networks applied to sky images for short-term solar irradiance forecasting. In *https://arxiv.org/abs/2005.11246*, 2020. 3

[9] Anto Ryu, Masakazu Ito, Hideo Ishii, and Yasuhiro Hayashi. Preliminary analysis of short-term solar irradiance forecasting by using total-sky imager and convolutional neural network. In *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, pages 627–631, 2019. 3

[10] Wanyi Xie, Dong Liu, Ming Yang, Shaoqing Chen, Benge Wang, Zhenzhu Wang, Yingwei Xia, Yong Liu, Yiren Wang, and Chaofang Zhang. SegCloud: a novel cloud image segmentation model using deep Convolutional Neural Network for ground-based all-sky-view camera observation. *Atmospheric Measurement techniques*, 2019. 7, 8