# Team GKN Project Report: GOES Satellite Cloud Detection, Tracking, and Prediction

Jonathan Kim, Kyler Norton, Andrew Geckle

ENEE439D: Topics in Signal Processing; Design Experience in Machine Learning
Professor Min Wu

**Table of Contents**

# Signature and approval

I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination.

Jonathan Kim, Andrew Geckle, Kyler Norton

Contributions: Project Report Creation, Preprocessing approaches and visualization techniques in Google Earth Engine/Google Colab, Development of ConvLSTM Model, Preliminary Research in Autoregressive and LSTM models.

# Executive Summary

Our project is centered around two major goals: image interpolation to fill in the 15 minutes of missing information between GOES snapshots and image prediction to take current cloud masks to produce potential prediction for cloud masks up to an hour in advance. The idea is that with both of these goals accomplished, More important to our project's goals are the potential benefits continuous and detailed data can offer in meteorologists' research. By developing models that can generate high-resolution predictions of cloud states between the taken snapshots, it is possible to enhance the granularity of available data. This can not only improve the accuracy of immediate weather forecasts but also aid in tropical storm tracking capabilities by using the in-between learnings as features in our predictive model.

To accomplish these tasks, we initially implemented base autoregressive and LSTM models. These simple models were unusable because they struggled with the 2D image cloud mask inputs we were passing in; such input required models designed to process spatial data. To address these issues, we developed a Convolutional Neural Network (CNN) which achieved moderate success by effectively handling the spatial aspects of the imagery. For our current validation and testing plans, we are simply subtracting the predicted cloud mask from the actual cloud mask in the testing dataset and comparing visually the amount of differences. In the future, it is advisable to take the difference mask and quantify as a percentage the amount of the screen is not blacked out, i.e. the amount of cloud that was either not caught or overpredicted by the model.

In order to further refine our approach, we made the decision to integrate the LSTM with convolutional layers to create a ConvLSTM model. This hybrid model harnessed both the spatial recognition capabilities of CNNs alongside the temporal/sequential data processing power of LSTMs, leading to improved performance in some scenarios. Simulation results indicated that the ConvLSTM model improved prediction accuracies significantly in the initial hours but showed declining accuracy in longer-term forecasts, suggesting a need for future models to incorporate more dynamic feature sets beyond simple spatial data.

The next big step for improving these cloud prediction models will focus on introducing more diverse feature sets since as of now, our model only looks at the shape of the cloud for learning. These potential feature sets to consider in future iterations include but are not limited to wind speeds, cloud top temperatures, and cloud particle densities. Future iterations should also focus on broadening the input features to include more specific atmospheric conditions pertinent to hurricanes in order to aid in tropical storm classification requiring training on datasets specifically involving tropical storms. The performance of our CNN cloud state interpolation, CNN cloud mask prediction, and LSTMConv cloud mask prediction prototypes have demonstrated that machine learning can substantially contribute to the field of meteorology specifically when it comes to filling in the missing data between cloud snapshots and also cloud predictions.

# Introduction

Our capstone project is designated to enhance the accuracy and timeliness of cloud tracking and prediction using geostationary satellite imagery. Geostationary Operational Environmental Satellites (GOES) are operated by the National Oceanic and Atmospheric Administration NOAA, which works to understand and predict changes in climate, weather, oceans, and coasts. These satellites provide continuous monitoring of atmospheric conditions over North America, and our project primarily uses data from GOES-16 with images taken 15 minutes apart from one another. The project employs advanced machine learning algorithms to analyze data from NOAA's GOES R satellite, which offers a rich dataset that captures various spectral bands alongside detailed thermal and cloud compositional data.

The end-users of this project will likely be meteorological agencies, disaster prevention meteorological organizations such as NOAA's National Hurricane Center, and the general population in hurricane-prone areas. By providing earlier and more precise warnings on the hurricane's path, we aim to significantly improve the preparedness and response strategies for tropical storms. Cloud classification and masking in our model was inspired from Deep Learning Models for Cloud Detection for Landsat-8 and Sentinel-2 images created through the international ESA-NASA collaborative.

## Objective

In response to the urgent need for enhanced predictive models in meteorology, we have chosen to develop an advanced system for hurricane cloud detection and path prediction utilizing geostationary satellite imagery. Our motivation for this project was to create a more accurate hurricane tracking model to ensure that disaster areas being affected by tropical storms could be warned in advance about the hurricane's path. As background, we are utilizing several satellites and their respective datasets to teach our model's cloud masking and hurricane tracking capabilities. These would likely include the NOAA, Sentinel-2, and Landsat 8 satellites with possibly more in the future depending on our model's learning requirements. These instruments provide us with a rich tapestry of data ranging from thermal signatures to cloud composition, all of which would distinguish and track hurricane patterns with unprecedented precision.

With this project, we aspire to transcend traditional methods by implementing machine learning algorithms that learn from historical datasets. Through the use of motion analysis, we plan to create backgrounds from frames across different days of the year grouped according to features such as time of day and season. Such methods would essentially be the incorporation of computer vision into our cloud tracking and learning literature. By combining already established scientific research in cloud detection alongside machine learning computational techniques, we hope to create accurate predictions for tropical storms that can be useful in emergency scenarios.

# Related/Prior Art

A geostationary satellite is launched to an altitude of about 22,000 miles above the equator, creating an orbit that matches the speed at which the planet rotates. These satellites are able to sense electromagnetic energy on different wavelengths (bands), each falling into one of three categories:

1. Visible
2. Near-Infrared
3. Infrared

While searching for methods used to identify/map clouds in images, we came across examples which utilize the open-sourced toolbox developed for sentinel 2 and the google earth engine. One such project was written by Minh Nguyen, the results of which can be seen in fig 1. These examples will be useful in the development of our own mapping algorithms, the only difference being that we will be selectively mapping tropical storms. Being able to identify and map onto the selected cloud formations will allow us to train our model on storm behavior and have it predict missing and future data.

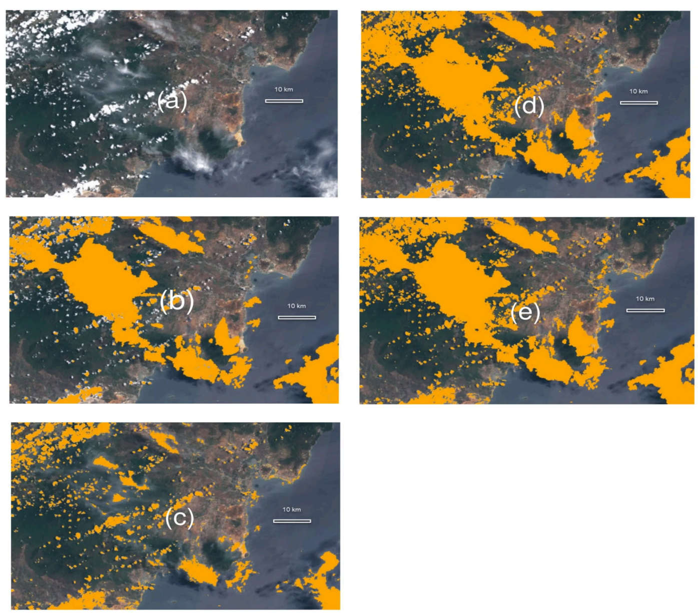



Figure 1: Sentinel 2 cloud mapping over google earth engine.

The topic of background subtraction as a method to mask clouds was brought to our attention by our professor and TA's. While looking into the possibility, we found a method proposed by Satrughan Kumar which uses background modeling to extract the foreground, and tracks the objects using Kalam filtering. To extract the foreground, initial empty frames are averaged together to form a reference background. Unless the work has already been done and is publicly available, in order to perform this method of extraction and tracking, we would need to stitch together cloudless parts of many images to build our initial frames. We would also need to create reference photos for the many different times of day and for each season.

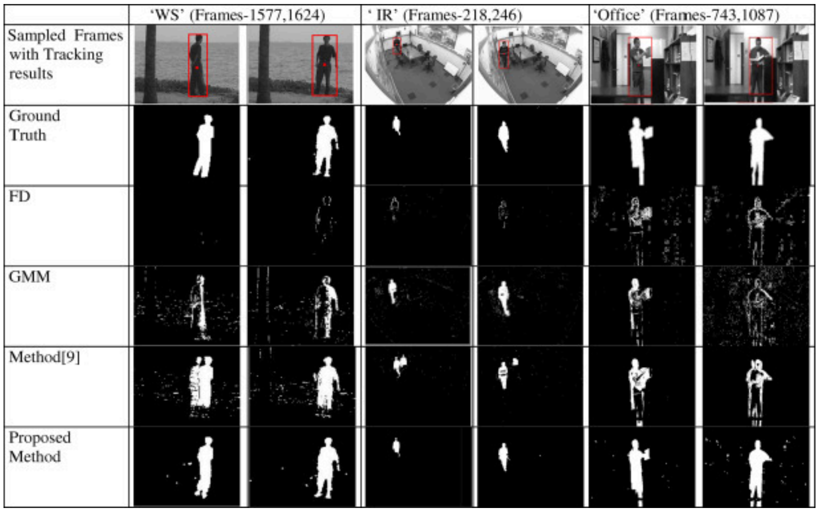



Figure 2: Satrughan Kumar's visual comparison of foreground motion masks

In order to distinguish tropical storms from other weather patterns, we learned that hurricanes can be easily identified by their shapes and temperatures. Hurricanes take shapes from ovals to circles (the stronger they are, the more symmetrical they become). The eye of a hurricane is the point about which the storm rotates, this area is another key identifier of a hurricane and its intensity. At a certain strength in cyclones, due to the coriolis effect, the eye is empty of clouds and the area experiences relatively mild conditions. Additionally, the temperature of cloud formations can be used to identify a hurricane. An inner ring of cold cloud temperatures which progressively get warmer as you move away from the eye is a distinctive feature of cyclonic weather.One way which we would be able to obtain temperature data on cloud formations would be by using the GOES R ABI band 11 which shows cloud top temperatures as seen in fig 3.

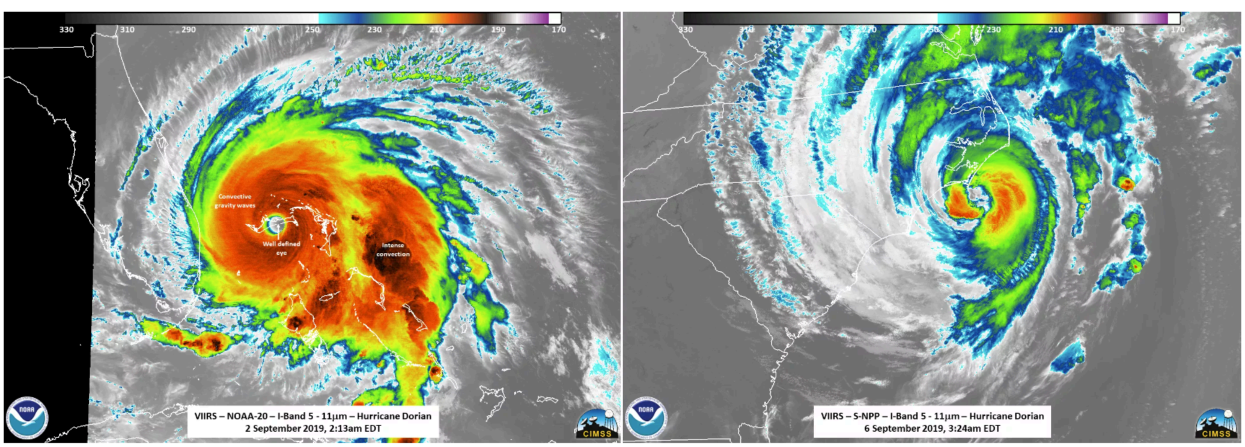
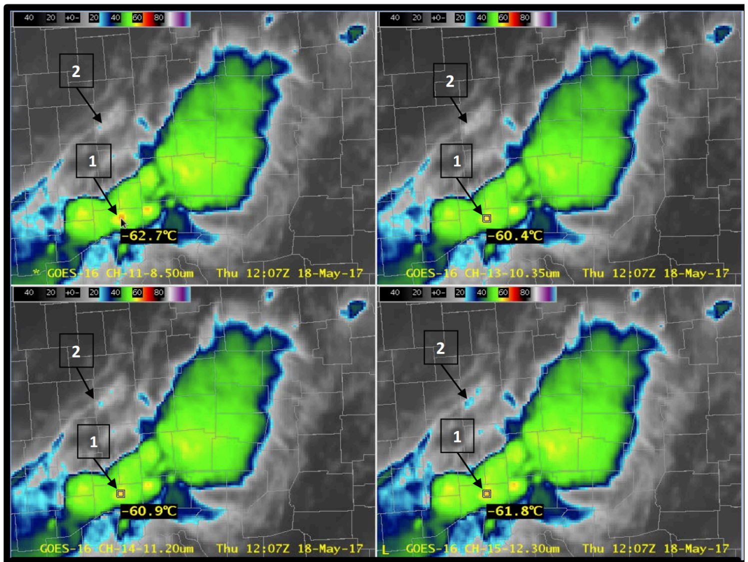



Figure 2: Images and temperature readings of Hurricane Dorian on 09/02/2019 and 09/06/2019

One way which we would be able to obtain temperature data on cloud formations would be through using the GOES R ABI band 11 which depicts cloud top temperatures as seen in fig 3.




*GOES-16 Window Channels, 18 May 2017 at 1207 UTC. Upper Left: 8.5 μm ; Upper Right: 10.3 μm ; Lower Left: 11.2 μm ; Lower Right 12.3 μm*

Figure 3: GOES R ABI band 11 imagery from provided documentation

The three datasets we are currently focusing our attention on are NOAA's GOES R, ESA's Sentinel 2, and NASA's Landsat 8 databases. Each of these programs launched numerous satellites which have been capturing imagery across multiple spectral bands for several years.

# Goals and Design Overview

Describe the project goals and design specifications, and why/how these goals and specifications have been modified over time, if appropriate. Provide an overview of the design, including a block diagram, key design sub-sections, descriptions of their functions, and how they relate/connect to one another. Outline the basic challenges that must be addressed. Provide a table of quantitative specifications for the overall project and for each component.

# Realistic Constraints

Identify and discuss at least five realistic constraints on the problem. Such constraints can include (but are not limited to) the following factors: Economic; Environmental; Social; Political; Ethical; Health and safety; Manufacturability; Sustainability; Legal; Regulatory and policy issues.

1. Government Regulatory Compliance: Must adhere to government and international standards on data handling and processing. Has to comply with any regulations concerning the use of satellite data since we are using NOAA's GOES-16 imagery. In order to make our model usable for government agencies, it is important that the accuracy of our model as well as its timeliness of weather prediction must be satisfied for operational use. This would include present-time model predictions we plan to

implement, something that would be critical for agencies such as the U.S Federal Emergency Management Agency to have up-to-date model accurate predictions on current tropical storm disasters.
2. Tourism - Economics: Tourism industry in America could benefit from more accurate weather analysis that could come from our ML model. As a result, companies such as Expedia would be interested in a ML model that is predictive not only minutes in advance, but perhaps days in advance as well. Our model should work to make sure that its predictive nature is viable for as far into the future as possible.
3. Insurance - Economics: Providing better risk assessment tools to utilize predictive weather modeling to forecast areas of high risk for weather-related claims is already a common practice. By improving such risk assessment through ML models learning through data specific to their regions of interest, we can offer better policies and premiums for companies such as State Farm. To do so, our model needs to be able to take as input images from different areas and be localizable.
4. Energy Sector - Predictions of cloud over and storm events for grid management and renewable energy sources are important for powering our country. In this way, both the government as well as individual companies in the energy sector will have great interest in a ML prediction model. This once in ties into the localizability issues that we seek to address with our model.
5. Agricultural Sector - Irrigation planning and risk assessment of weather impacts on crop yields. Model could serve to provide specialized agricultural forecasts that include not only cloud movements but also predictions about precipitation, temperature, and other climatic factors affecting agriculture. For this reason, it would be good for our model to later expand on its original premise of simple cloud tracking and extend to track these other features also offered by GOES-16 across different bands.

# Engineering Standards

IEEE 2674-2019: Standard for machine Learning Process Framework and Lifecycle
When it comes to the maintenance of our model across different dates, it is important that some form of automated testing and validation procedures are present. The framework for automated validation is accomplished by quantifying the difference between the predictions of the model against the actual test images; if the model begins creating predictions which are too different (10%+ more than actual image) then it is possible to temporarily retire the system until maintenance.

IEEE 830-1998: Recommended Practice for Software Requirements Specifications
Ensuring that all software components of the hurricane system are clearly defined, understood, and agreed upon. All members of the group discussed potential model choices and major decisions such as the choice to switch from Google Earth Engine to Google Colab as a virtual environment. From there, all software created was done so with full knowledge from the entire group as to its purpose. One file was discussed to be created for loading in images into a dataset, another file was discussed to be specific to image interpolation, etc.

IEEE P7001: Transparency of Autonomous Systems
This standard is very relevant for our project since it involves machine learning. It is our duty to address transparency in how the autonomous system makes decisions. In our case, we offer extensive documentation both through reports not unlike this one and also presentations. In these reports and presentations, the logic behind our machine learning models is made clear which is important for regulatory compliance.

# Design Choices

## Chosen Cloud Tracking: LSTMConvolutional Neural Networks

The convolutional LSTM model represents a fusion of CNNs and LSTM networks designed to tackle the challenges of spatio-temporal data processing. Our group decided this model is theoretically the best-suited for cloud tracking applications to handle the spatial relationships (edge detection and movement) as well as the temporal relationships (cloud movements across different times of year or times of day).

Pros:
- Spatio-Temporal Processing which is necessary for tracking the cloud's shapes over temporal sequences leading to better performance in short-term predictions

Cons:
- Relatively low available literature on existing LSTMConv models compared to the other options, few benchmark options for comparison
- More complex to implement and fine-tune compared to using a basic CNN
- Longer training times
- Significantly more computational resources, require higher GPU and CPU memory caps in Google Colab (A100 premium GPU necessary)

## Chosen Cloud Detection: Background Subtraction

Background subtraction was eventually our chosen method of masking the clouds as input into the CNN models. This method involves creating a model of the background and subtracting it from the current image to highlight changes, which in this case would be moving clouds. This is important in detecting new cloud formations appearing over short periods since there is not exact indicator or patterns of clouds forming spontaneously without looking at extraneous factors such as water particle density or temperatures. Our original iteration looked to try to create a dynamic background that accommodated for the different lighting present throughout the day to use for background subtraction. However, we found a much simpler alternative in band applications. Our method was to combine specific visual bands to create a layer that had the full functionality of a cloud mask, only showing the clouds. In this way, the only work that had to be done was thresholding the values of those three bands in a way that all non-blue portions of the map (being the landmasses or sea) would be completely removed and all that would remain are the clouds.

Pros:

- Very simple to implement since it is almost completely accomplished through visual banding.
- Execution of the background subtraction takes little to no time

Cons:

- Static background limitation: Assumes a relatively static background which would be problematic for RGB applications without the visual banding. This method of background subtraction is only applicable to banded images
- Does not distinguish between different types of clouds, something which may be useful for more advanced models that could look at cloud type as a possible input feature to the model.

## Alternative Cloud Detection: Object Detection

Mask R-CNN extends Faster R-CNN by adding a branch for predicting segmentation masks on each Region of Interest to allow precise object segmentation. In this case, we would be precisely segmenting cloud formations. However, it was not chosen due to it being both computationally intensive and also much harder in implementation when compared to our chosen strategy of masking the background through different visual bands to easily extract clouds.

Pros:

- Allows for precise segmentation of cloud formations that provide detailed and accurate cloud masks
- Capable of handling complex cloud scenes and differentiating between the different cloud types.

Cons:

- Computationally intensive and resource-demanding, which would significantly hinder our model creation progress
- Slower in image processing compared to the simpler object detection

## Alternative Cloud Tracking: Autoregressive Models

Autoregressive models predict future behavior based on past values, making them very useful in time-series analysis. We initially believed that such models could be applied to cloud tracking by modeling the next state of the clouds based on previous observations. This model was a prime alternative because of its simplicity and effectiveness in cases with linear relationships and short-term dependencies. However, we found that AR models struggle with complex patterns and long-term dependencies, specifically in capturing spatial dependencies needed for cloud predictions. Understanding the parts of the image and how they spatially relate to one another is important for clouds which shift, amalgamate, and split from each other. As such, Autoregressive models appear to be insubstantial for our purposes.

Pros:

- Simple to understand and implement, effective in cases with linear relationships with short-term dependencies
- Much lower computational power required for training

Cons:

- Struggles with complex patterns and long-term dependencies, both of which are common in meteorological data
- Ineffective for nonlinear data relationships which are present in weather patterns
- Unable to capture spatial dependencies desired in image prediction and analysis.

## Alternative Cloud Tracking: LSTM Models

LSTMs are a type of recurrent neural network designed to learn long-term dependencies. While LSTMs are highly effective for sequence prediction and time-series forecasting, they face the same issue that autoregressive models face in that they are unable to identify spatial relationships. As such, these models were insubstantial for our purposes.

Pros:

- Effective at learning long-term dependencies in time-series forecasting. Suitable for predicting sequences over time which could be attributable to climates or regional weather patterns.

Cons:

- Cannot capture spatial dependencies desired in image prediction and analysis.
- More complex to train compared to simpler autoregressive models

## Alternative Cloud Tracking: Convolutional Neural Networks

Convolutional Neural Networks are highly effective for image processing tasks due to their ability to automatically detect and utilize spatial hierarchies in data. Looking at many other benchmark models we based our work off of in the field of image processing, CNNs are the primary choice. For both image interpolation between different frames as well as image prediction, CNNs are seen to excel at learning the general edges of the clouds. This edge detection is crucial since it is the most important (and frankly only) feature being learned through training epochs. However, they are unable to process temporal information which means it will be unable to identify seasonal or even climatic behavior of the clouds. This leaves such a model as the second most desirable model for cloud prediction.

Pros:

- The most common option for image processing tasks due to its ability to detect spatial hierarchies in data.

- Excels at learning and detecting edges and shapes, both of which are important for identifying cloud formations
- Abundance of literature on existing convolutional neural networks.

Cons:

- Unable to process temporal information, making it very limited in how far predictions can see into the future. Also makes the model less accurate since we can't learn climatic behaviors.

# Technical Analysis for ConvLSTM

The ConvLSTM extends the traditional LSTM by replacing the fully connected layers with convolutional layers. The key equations of ConvLSTM are shown below, where * denotes the convolution operator and the circle representing the Hadamard product:

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \odot \mathcal{C}_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \odot \mathcal{C}_{t-1} + b_f)$$

$$\mathcal{C}_t = f_t \odot \mathcal{C}_{t-1} + i_t \odot \tanh(W_{xc} * X_t + W_{hc} * \mathcal{H}_{t-1} + b_c)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \odot \mathcal{C}_t + b_o)$$

$$\mathcal{H}_t = o_t \odot \tanh(C_t)$$

The first line describes the input gate, which controls how much of the new input is written to the cell state. $W_{xi}$ * $X_t$ is the convolution of the input $X_t$ with the weight matrix $W_{xi}$. $W_{hi}$ * $H_{t-1}$ is the convolution of the previous hidden state with the weight matrix. $W_{ci} \circ C_{t-1}$ is the element-wise multiplication of the previous cell state with the weight matrix $W_{ci}$ and $b_i$ represents the bias term.

The second line describes the forget gate, which determines how much of the previous cell state should be forgotten. $W_{xf}$ * $X_t$ is the convolution of the input $X_t$ with the weight matrix $W_{xf}$. $W_{hf}$ * $H_{t-1}$ is the convolution of the previous hidden state $H_{t-1}$ with the weight matrix $W_{hf}$. $W_{cf} \circ C_{t-1}$ is the element-wise multiplication of the previous cell state $C_{t-1}$ with the weight matrix $W_{cf}$.. $b_f$ represents the bias term.

The third line describes the cell state update, which combines the previous cell state and the new candidate values. The first term $f_t \circ C_{t-1}$ is the element-wise multiplication of the forget gate output ft with the previous cell state $C_{t-1}$ indicating what to remember from the past. The second term it ◦ tanh($W_{xc}$ * $X_t$ + $W_{hc}$ * $H_{t-1}$ + bc) represents the new candidate values to be added to the cell state, where $W_{xc}$ * $X_t$ is the convolution of the input with the weight matrix. $W_{hc}$ * $H_{t-1}$ is the convolution of the previous hidden state with the weight matrix, and the tanh function normalizes values to be between -1 and 1.

The fourth line is the output gate, which decides how much of the cell state is output to the hidden state. $W_{xo} * X_t$ is the convolution of the input Xt with the weight matrix $W_{xo}$, $W_{ho} * H_{t-1}$ is the convolution of the previous hidden state with the same weight matrix. $W_{co} \circ C_t$ is element-wise multiplication of the current cell state $C_t$ with the weight matrix.

The fifth line is the computation of the final new hidden state which is used in the next time step and also as output. $H_t = o_t \circ \tanh(C_t)$ is the element-wise multiplication of the output gate ot with the hyperbolic tangent of the current cell state $C_t$. This equation allows the model to continuously relegate new information through the time steps.

To reproduce the ConvLSTm model for cloud tracking, the following steps must be taken:

1. Data collection - Acquire satellite images from NOAA's GOES-16 satellite and take note of ther imageID to ensure they can be used in the NewConvLSTM google colab file.
2. Apply the specified image bands to ensure that the cloud mask is properly created.
3. Normalize pixel values of the imported images to a range of [-1, 1] for sigmoid activation.
4. Split the dataset into training and test sets. For our purposes, a 80-20 split was done.
5. Model Architecture - Utilize three ConvLSTM2D layers with a batch normalization layer between each to reduce processing time and improve efficiency. The first layer will be done by 5x5 kernels to ensure that we capture the spatial features related to big cloud movements. The second ConvLSTM2D layer will incorporate 3x3 kernels to capture the finer details involving cloud edges. The last ConvLSTM2D layer incorporates 1x1 kernels simply to provide easier input into our final layer. Our final output layer will be a 3D layer containing the weights necessary for our prediction.
6. Train the model using the training dataset using a binary cross entropy loss function to validate its performance.
7. Evaluate the final model on the test set to assess its performance, the methods towards evaluation of our model are described below.

Based on our results, we have determined that experimenting with different filter sizes, number of filters, learning rates, and batch sizes were essential to ensure that model training alone was feasible. Afterwards, model complexity was taken into account. The number of ConvLSTM was actually reduced from 4 to 3 due to computational difficulties. Perhaps future designs could look towards 4 ConvLSTM layers as long as overfitting is taken into account. Incorporating additional meteorological features such as wind speed and cloud particle are of utmost importance in future iterations due to our model's difficulty in determining whether a cloud is about to dissipate or appear on the screen. By far the largest shortcoming of our model is its computationally intensive nature, requiring hours of significant computational resources and time for both training and inference. When it comes to performance, the only major issue is the failure to predict cloud spontaneous dissipation and formation. The issue with our model comes from our incorrect base assumption that cloud formation and dissipation can be generalized to a pattern off of shape alone, which is obviously not the case. Such information must be gathered from extraneous features that we have not explored in this venture as of yet, however my first recommendation would be looking into cloud particles. Cloud particles represent the prevalence of aerosol particles in a region which are necessary building blocks for the initial formation of clouds. As such, introducing new weights trained on cloud particle movement visual bands could be essential for a fully functional model in the future.

# Design Validation for System and Subsystems

Our LSTM model uses five sequential input images at fifteen minute intervals and predicts the next image in said sequence. Below are the results of testing done on its ability to predict a single frame of data, and to make consecutive predictions with previous outputs being used as new input data.

## LSTM Single Frame Prediction

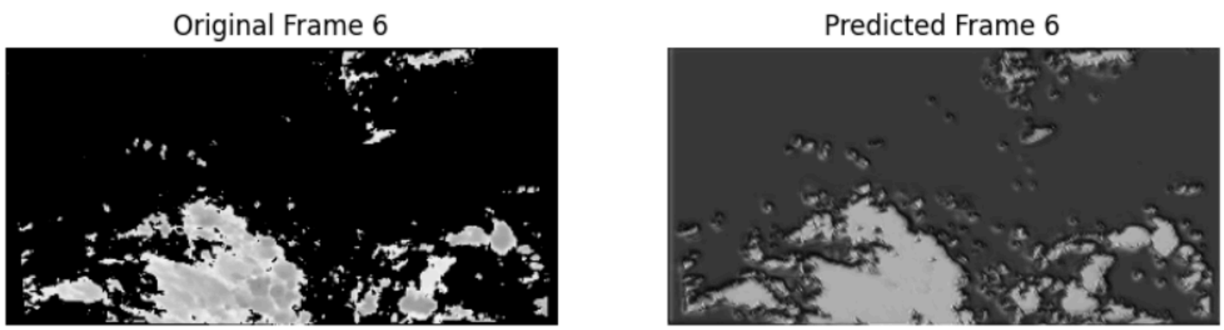



Figure 4: LSTM predicted frame compared to the true data.

The LSTM model is capable of predicting the movement of large cloud bodies, but has trouble with the movements and formations of smaller clouds. In the figure above, it can be seen that there are multiple small clouds predicted that do not exist in the label data.

## LSTM Sequence of Predictions

The below series of images are a sequence of five consecutive predictions made by the LSTM model as well as the ground truth images obtained from the GOES-R satellite.

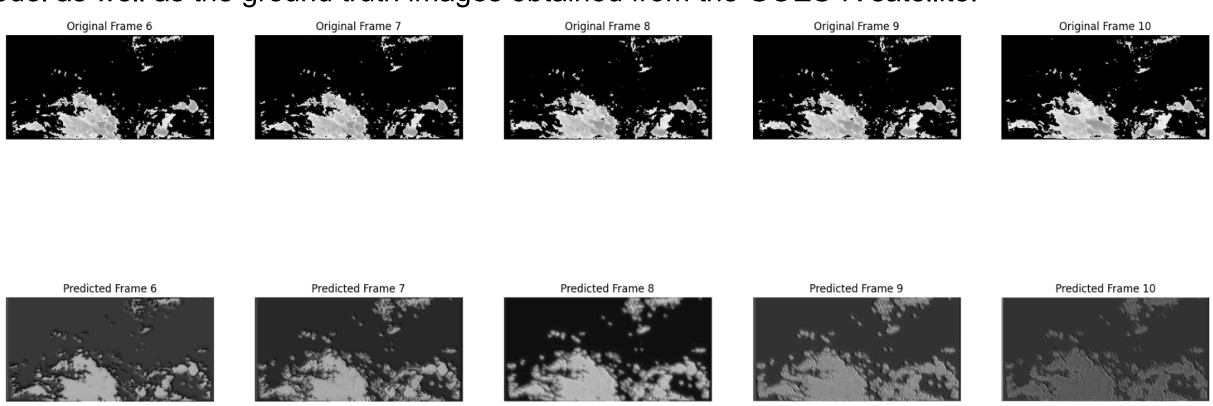


Figure 5: Sequence of five LSTM predictions compared to the true data.

The model tends to stretch existing cloud formations. As more predictions are made one after the other, the cloud structures begin to expand and form into a single mass.

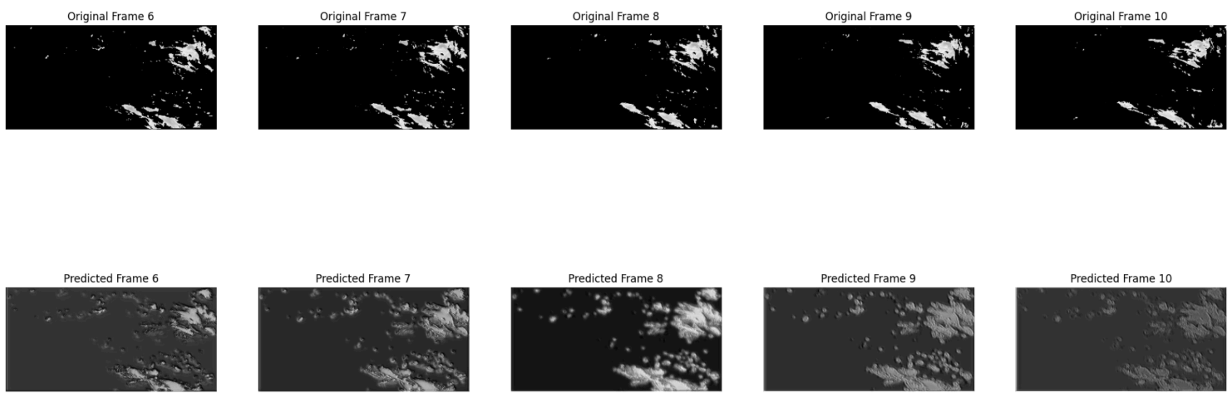



Figure 6: Sequence of five LSTM predictions compared to the true data.

Fig. 6 is another great example of the model arbitrarily adding new clouds to the output when none are supposed to appear according to the labels. This source of error can likely be reduced by introducing new banding information into the input images which concern atmospheric water vapors. Such data should allow the model to foresee the development of new clouds and refrain from adding them arbitrarily.

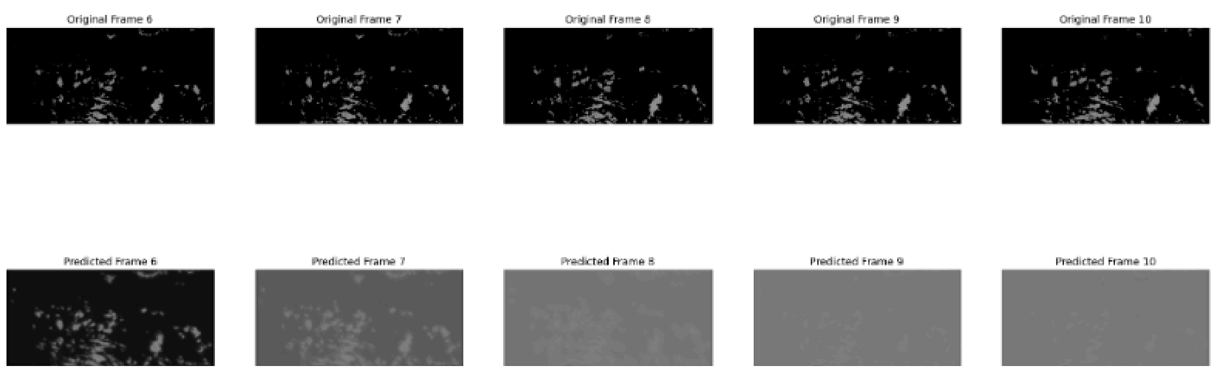



Figure 7: Sequence of five LSTM predictions compared to the true data.

Another issue faced by the LSTM model is that of predictions becoming gradually brighter under some circumstances. This is believed to be due to gradient vanishing, and can potentially be solved using different activation functions or by processing the outputs between predictions to reduce the effects.

Without doubt, there are still many holes to patch in the model. However, the ability for the model to predict cloud movement with decent accuracy when concerning larger cloud bodies indicates potential usefulness when predicting the pathing of hurricanes.

# Test Plan and Overall Performance Achieved

The design will be evaluated by taking the image mask differential between the predicted image and the test set image. Since our dataset is split 80/20 respectively into the training dataset and the testing dataset, to assess whether the design is working properly we have two options.

The first option will be to take random sequences of images from the testing data set and run the model on the first image. From there, we can compare the next 3-4 predicted images output from the model with the remaining images in the test sequence. Afterwards, we can subtract the predicted cloud mask with the actual cloud mask as a different cloud mask. From here, we find the percentage of the mask that is not black indicating an incorrect prediction at that location. This percentage will represent the performance accuracy of our model. A completely perfect prediction will have no difference between the predicted mask and the actual mask resulting in a difference mask that is completely black, 100% accuracy.

The second approach to measuring the performance of the model is to take live NOAA GOES satellite imagery over our specified coordinates in the Gulf of Mexico as input into our model directly. Quantifying the accuracy will be the same as the previous approach.

# Project Planning and Management

Project management organization and responsibilities was described in a group setting, with responsibilities being allocated to individuals by the group as a whole. For the initial preprocessing steps of our capstone project, Kyler dealt with band testing to visualize the effects of different band combinations to determine which combinations were best at extracting specifically clouds from the satellite imagery, as well as cloud masking techniques. Jonathan was tasked initially with band research, later moving into visualization techniques, cloud masking techniques, and exporting the Google Earth image layer cloud masks into Google Drive. A.J was tasked with determining regions of interest as well as doing preliminary research into benchmark machine learning powered cloud detection models.

The machine learning steps of our project were divided into the following: AJ researched the possibility of optical flow as a layer into the CNN and ConvLSTM models to enhance accuracy in addition to assisting Kyler in the creation of the functional CNN model. Kyler worked on the bulk of the creation of the CNN predictive and interpolation model. Jonathan worked on the creation of the ConvLSTM model as well as some preliminary investigations as to the possibilities of Autoregressive and LSTM model, in addition to some research on the image interpolation CNN model. Task breakdown and scheduling proceeded with little difficulty.

| **Task** | **Owner** | **3/18-3/24** | **3/25-3/31** | **4/1-4/7** | **4/8-4/14** | **4/15-4/21** | **4/22-4/28** | **4/29-5/5** | **5/06-5/12** |
|---|---|---|---|---|---|---|---|---|---|
| Data Collection | Jonathan | X | X | | | | | | |
| Cloud Mask using Background Subtraction | Jonathan | X | X | X | | | | | |
| Extract Features Relevant to Storms | Kyler | | X | X | X | | | | |
| ML Model Development | AJ | | | X | X | X | | | |
| Testing Prototype Model | AJ | | | | X | X | | | |
| Documentation of Findings | Kyler | X | X | X | X | X | | | |
| First Presentation | All | | | | X | X | | | |
| Updating/Reworking Models | All | | | | | X | X | X | |
| Final Report | All | | | | | | | X | X |

# Conclusions

Our project embarked on the ambitious goal of enhancing cloud tracking and prediction using machine learning algorithms applied to geostationary satellite imagery. Our team explored a variety of modeling techniques, beginning with our unique preprocessing approach. We did a form of pseudo-background subtraction where the GOES images were applied with a combination of imaging bands to produce a layer in the Google Earth Engine map displaying only the clouds present in the image. Afterwards, thanks to our choice of using Google Colab as our virtual environment, we are able to directly export this layer into Google Drive as a png in a folder representing our dataset of masked and unmasked images. From here, we focus more on applying a variety of different machine learning models and methodologies to accomplish our tasks. We came to the conclusion that a CNN was sufficient for image niterpolation in filling the missing information between GOES snapshots, but a ConvLSTM was preferable for satellite image prediction. This development of this hybrid ConvLSTM represents a significant advancement in the field of cloud analysis due to the majority of other benchmark models relying on base CNN predictions. We believe that a ConvLSTM more effectively captures the dynamics of cloud movement specific to the climate or season by keeping a temporal memory of the behavior of clouds. Additionally, our usage of thresholding through different visual bands to create a comprehensive cloud mask over the traditional background subtraction is effective at reducing preprocessing complexity and time cost.

The potential impact of our project is considerable since even the based model with only spatial features being considered is able to provide somewhat accurate and timely predictions of cloud movements, something that can significantly benefit weather forecasting and climate research. If our project was to receive additional time and resources, it is possible to increase the features being learned by the model instead of simply looking at cloud position and edges. Such a model could have enhanced accuracy over our current iteration, consequently leading to better cloud interpolation and prediction. This has the possibility of saving lives through better disaster preparedness and making considerable strides in the environmental science sphere of research. The project was a considerable learning experience for the entire team. We had to learn that rigorous testing and validation is essential in developing any model to perform well in real-world scenarios. Our initial projections from the CNN and ConvLSTM were abysmal before we experimented with different input sizes, batch sizes/epochs, and model layers. To solve the majority of our issues required repetitive testing; in order to optimize cloud detection we had to iterate through countless combinations of visual bands with unique thresholds. In order to improve our model accuracy we had to repetitively retrain the model with different layers, batch sizes, and input image size constraints. Repetition was the key to the majority of the challenges we faced in this project. We also learned that different models serve drastically different purposes; it is incredibly important that one researches which model can accommodate the issue at hand.

In conclusion, our project has laid a strong foundation for machine learning and its relevance in the field of meteorology. We demonstrated that machine learning could play a pivotal role in predicting the skies above. More importantly, we demonstrated the room for growth such a project has for future groups. The possibility of incorporating not only more diverse datasets aside from the Gulf of Mexico but also investigating more sophisticated modeling techniques can enhance the accuracy and reliability of a ConvLSTM model.

# References

The text should cite at least two sources per team member, with at least half of the sources from peer-reviewed archival publications such as journal or conference papers. Use IEEE format for the citations and bibliography.

[1] A. Meraner, P. Ebel, X. X. Zhu, and M. Schmitt, “Cloud removal in sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 333–346, Aug. 2020. doi:10.1016/j.isprsjprs.2020.05.013

[2] GfG, “Deep learning: Introduction to long short term memory,” GeeksforGeeks, https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/ (accessed Mar. 15, 2024).

[3] “‘sentinel-2 cloudless mosaic’: The first global (almost) cloud-free view of the planet,” Copernicus, https://www.copernicus.eu/en/sentinel-2-cloudless-mosaic-first-global-almost-cloud-free-view-planet (accessed Mar. 15, 2024).

[4] Author links open overlay panelSatrughan Kumar and SummaryBackground subtraction is an efficient way to localize and obtain the centroid of the connected pixels moving on the foreground despite the prior information of the scene. It is suitable under fixed camera arrangement, “Video object extraction and its tracking using background subtraction in complex environments,” Perspectives in Science, https://www.sciencedirect.com/science/article/pii/S2213020916300866#:~:text=Background%20subtraction%20is%20an%20efficient,tracking%2C%20human%20monitoring%2C%20etc. (accessed Mar. 15, 2024).

[5] “Papers with code - SEN12MS-CR dataset,” Dataset | Papers With Code, https://paperswithcode.com/dataset/sen12ms-cr (accessed Mar. 15, 2024).

[6] “Sentinel-2 cloud masking with s2cloudless | Google Earth engine | google for developers,” Google, https://developers.google.com/earth-engine/tutorials/community/sentinel-2-s2cloudless#build_a_sentinel-2_collection (accessed Mar. 15, 2024).

[7] Sentinel-Hub by Sinergise, “Braaten-Cohen-Yang Cloud Detector,” Sentinel Hub custom scripts, https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/cby_cloud_detection/ (accessed Mar. 15, 2024).

[8] R. Awati, “What is a geostationary satellite?: Definition from TechTarget,” Mobile Computing, https://www.techtarget.com/searchmobilecomputing/definition/geostationary-satellite (accessed Mar. 15, 2024).

[9] B. Pflug, R. Richter, R. de los Reyes, and P. Reinartz, “Comparing atmospheric correction performance for sentinel-2 and landsat-8 data,” *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2019. doi:10.1109/igarss.2019.8900095

[10] “A guide to understanding satellite images of Hurricanes,” National Environmental Satellite, Data, and Information Service, https://www.nesdis.noaa.gov/news/guide-understanding-satellite-images-of-hurricanes (accessed Mar. 15, 2024).

[11] M. Nguyen, “Improve cloud detection and removal with machine learning in Google Earth engine,” Medium, https://medium.com/eelab/improve-cloud-detection-and-removal-with-machine-learning-in-google-earth-engine-ac0a2f759022 (accessed Mar. 15, 2024).

[12] M. Nguyen, O. Baez-Villanueva, D. Bui, P. Nguyen, and L. Ribbe, “Harmonization of Landsat and sentinel 2 for crop monitoring in drought prone areas: Case studies of Ninh Thuan (Vietnam) and Bekaa (Lebanon),” *Remote Sensing*, vol. 12, no. 2, p. 281, Jan. 2020. doi:10.3390/rs12020281