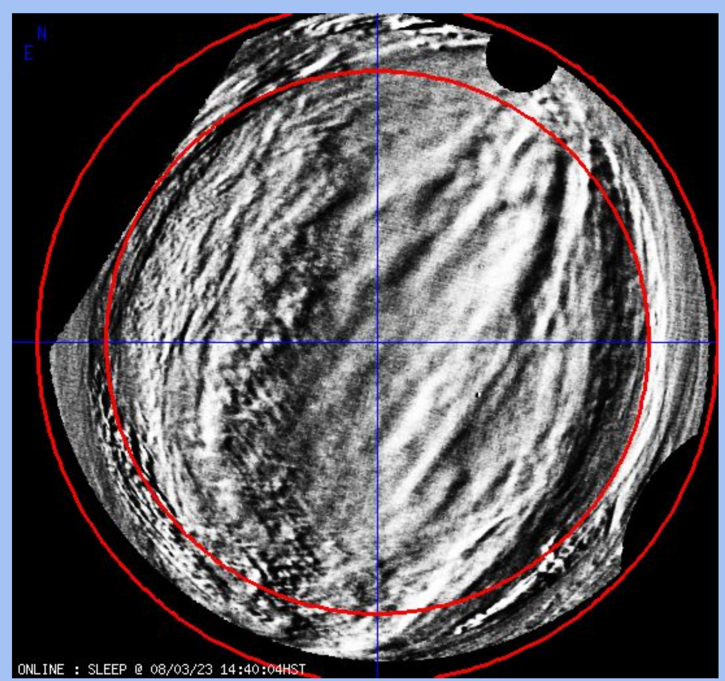
Using a convolutional neural network with all sky infrared images to classify sky regions as clear or cloudy

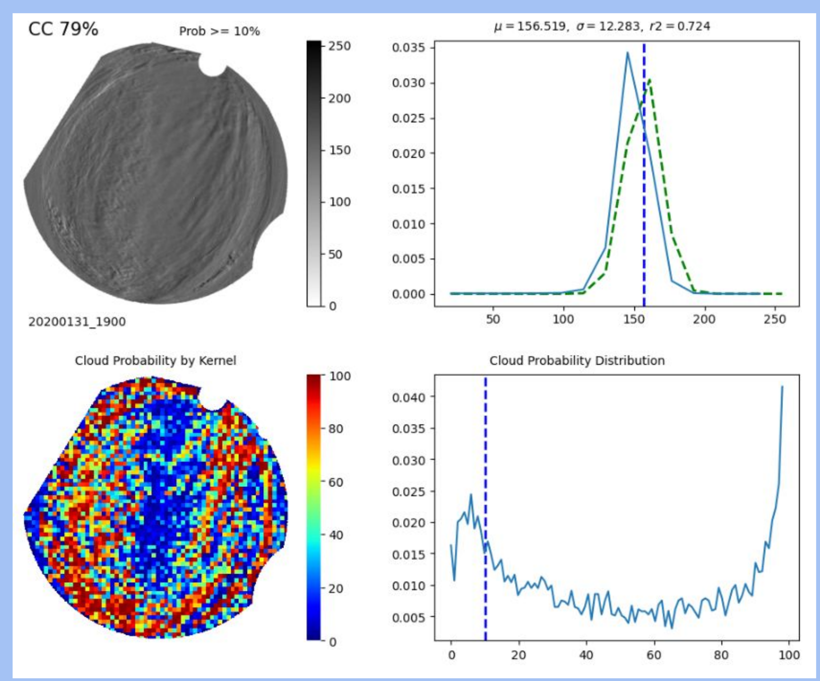
Brock Taylor
Canada France Hawaii Telescope
Columbia University: Fu Foundation School of Engineering and Applied Science
Mentor: Billy Mahoney

Abstract

Used a Convolutional Neural Network to detect clouds on Mauna Kea using Canada France Hawaii Telescope's (CFHT) All Sky Infrared and Visible Analyzer (ASIVA). Two models were constructed: a full-sky image classifier and a heatmap generator based on different size pixel kernels. The full-sky classifier was able to determine clear skies with 100% accuracy (0% false positive rate) and cloudy skies with 96% accuracy (4% false negative rate). The heatmap generator model used a machine learning network on a small kernel which it passed over an input image to determine the likelihood of cloud coverage at each location. Data cleaning was required to yield significant results due to dynamic range limitations of the sensor causing significant differences between clear and cloudy images. Different batch sizes were compared to test model convergence, ROC performance, and overall effectiveness. Smaller batch sizes were found to be more effective with a batch size of 32 yielding an AUC of 0.987. Cloud coverage percentage was determined by comparing each kernel's prediction value against a determined threshold constant and dividing the number of kernels classified as cloudy by the total number of kernels. Overall, the heatmap model was found to provide significantly more data than the current system on cloud coverage over Mauna Kea. Cloud coverage percentage is a metric not currently available for continuous image acquisitions over Mauna Kea. Additionally, the heatmap approach provides data on cloud coverage in specific sky regions, allowing for much more accurate observations of cloud activity in regions of interest.

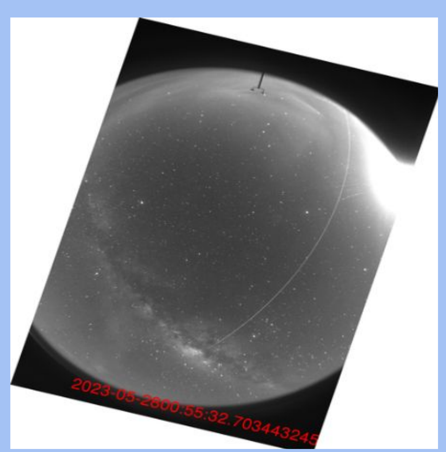


Current ASIVA Display

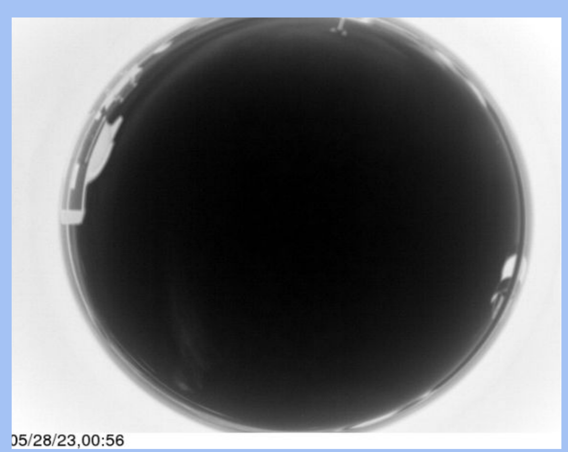


Heatmap Data Display

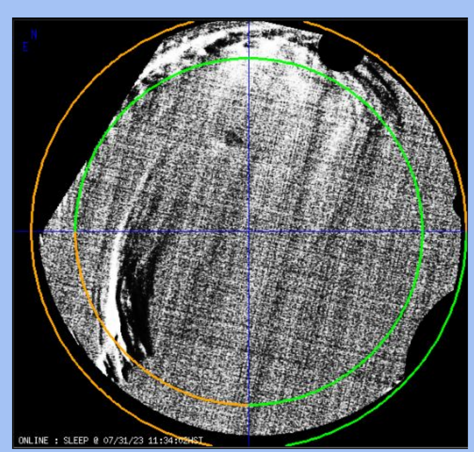
The ASIVA



Visible-Light Image



IR Image (Raw)



IR Image (Differential)

ASIVA Instrument

The ASIVA (All Sky Infrared and Visible Sky Analyser) is an Infrared and Visible camera system maintained by CFHT that uses a Support Vector Machine (SVM) to detect cloud coverage in 8 distinct sections of the sky on Mauna Kea. It was installed in 2010 and produces 2 raw images and 1 differential image every minute, meaning a database of around 2 million images is available. The Support Vector Machine used to analyze the differential images produced takes as input the parameters of a Gaussian fit for each of the 8 distinct regions, meaning 32 data points represent the full image. The SVM produces a numerical output for the level of cloud coverage in each region, and a separate script classifies each region based on that output as clear, cirrus, or cloudy. However, since it only classifies 8 regions of the sky, actual determination of where clouds are is done manually through visual inspection.

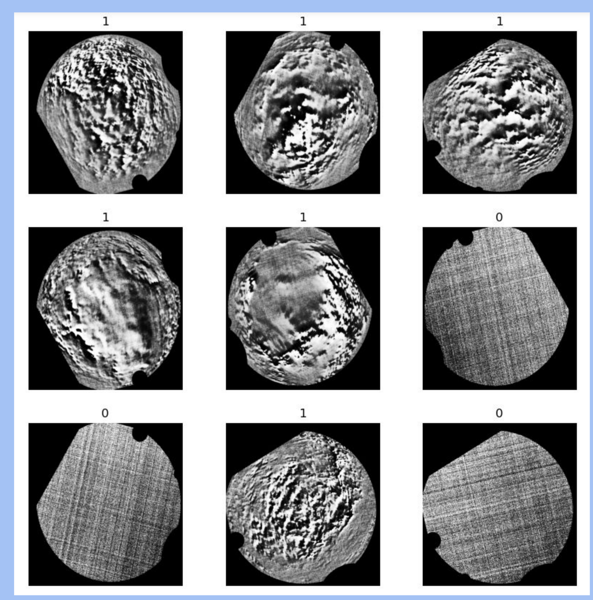
Project Goals

The goal of this project was to develop a program that could detect clouds on Mauna Kea with higher accuracy and resolution than the existing program used by ASIVA. To do this, two models were created: a full-sky image classifier and a heatmap generator program that splits the image into numerous kernels and classifies each kernel. Both of these models use the IR differential image dataset from the ASIVA as their inputs. As this was my first time implementing a machine learning program, the full-sky image classifier was made to familiarize myself both with the ASIVA dataset and with the tools available to me for developing such a program.

Building a Full-Sky Image Classifier

Model and Dataset

Full-sky IR images from the ASIVA were classified based on Gaussian fit parameters to construct the dataset. Clear images produced by the ASIVA are largely instrument noise. Therefore, a Gaussian curve fits this data well and can be used to identify clear skies. However, this analysis isn't perfect, and can often be obstructed by the presence of cirrus clouds which are sparsely covering sections of the sky or by dense clouds present only in certain sky sections. Because of this, manual data cleaning was done to parse these problematic images from the dataset. A more thorough analysis of the full-sky classifier model could be done to include these images as a third category of "cirrus" or a fourth category of "dense not covering" to more accurately represent real-time data produced by the ASIVA. However, as I wished to move onto the heatmap generator model, I was unable to further the analysis to include these image types due to time constraints. Simple image transformations were added to the dataset such as rotations and reflections. A Relatively simple model architecture was used to analyze pixel values from the images to produce a confidence level for classifying the images as cloudy.



Example ASIVA IR Images

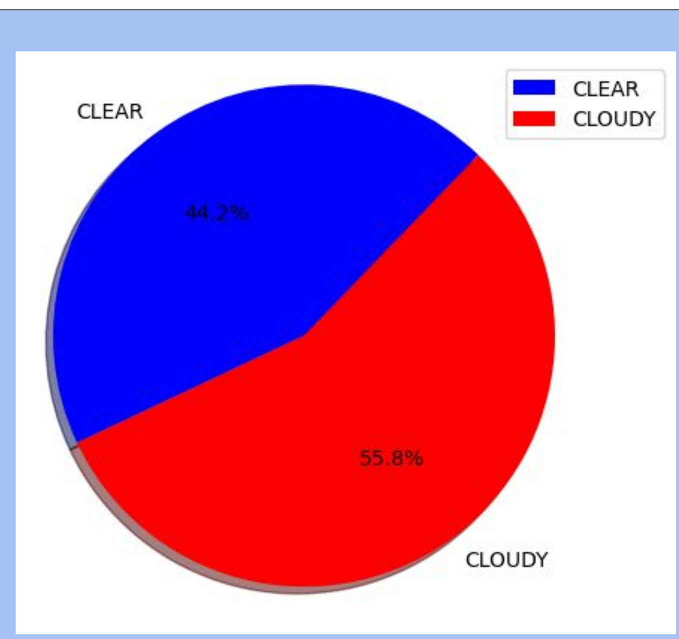


Image Class Distribution

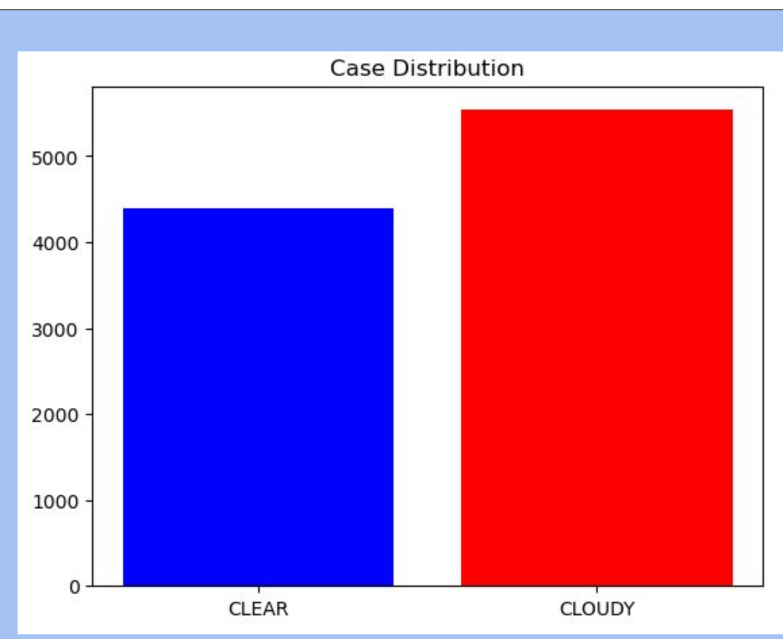
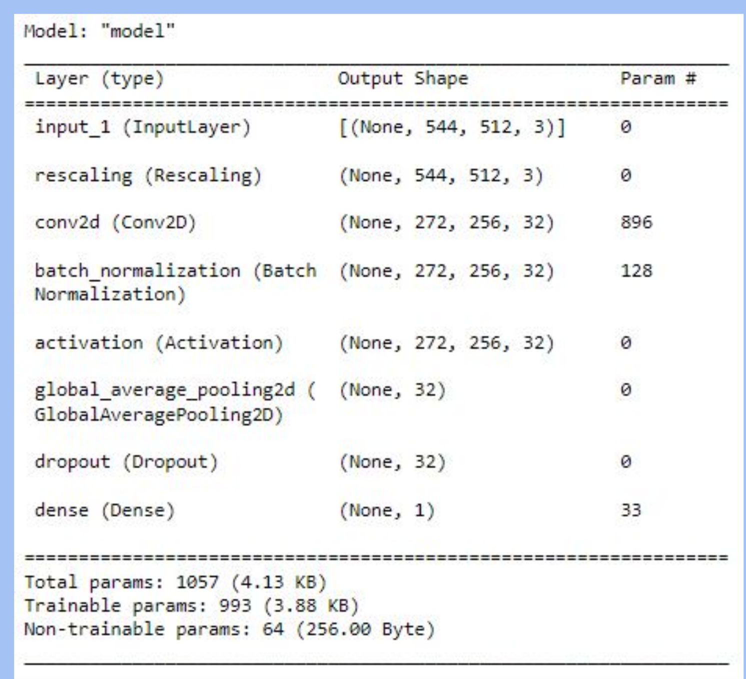


Image Class Distribution

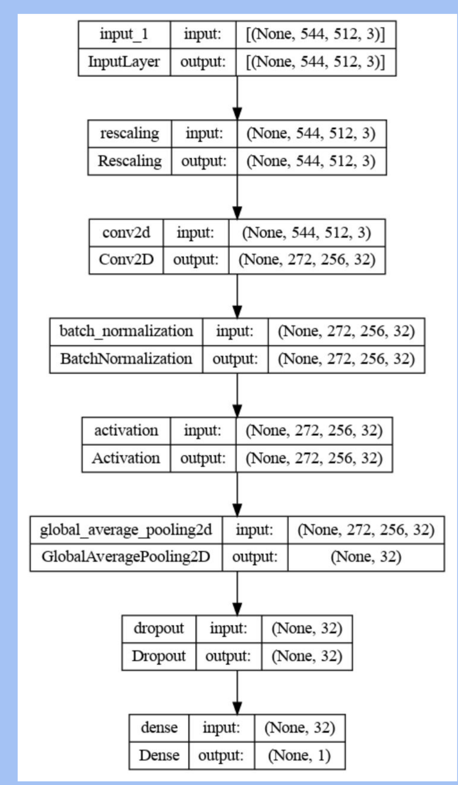


Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 544, 512, 3)]	0
rescaling (Rescaling)	(None, 544, 512, 3)	0
conv2d (Conv2D)	(None, 272, 256, 32)	896
batch_normalization (Batch Normalization)	(None, 272, 256, 32)	128
activation (Activation)	(None, 272, 256, 32)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 32)	0
dropout (Dropout)	(None, 32)	0
dense (Dense)	(None, 1)	33

Total params: 1057 (4.13 KB)
Trainable params: 993 (3.88 KB)
Non-trainable params: 64 (256.00 Byte)

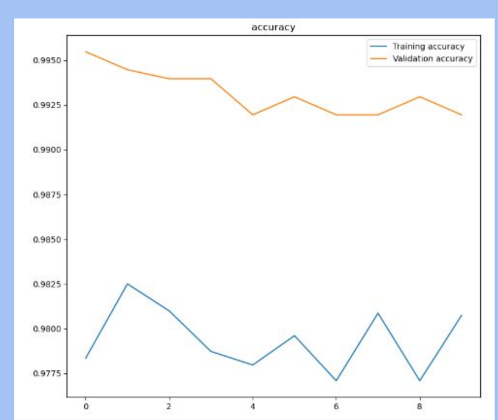
Model Architecture



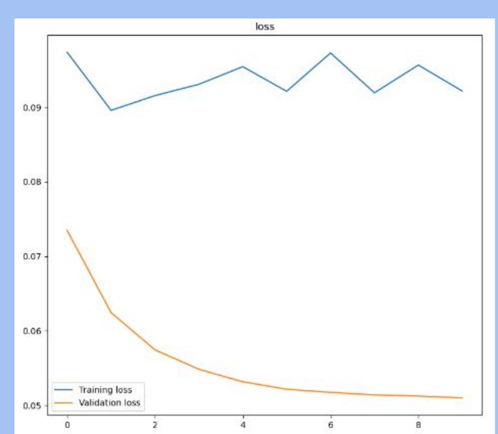
Model Architecture

Full-Sky Classifier Results

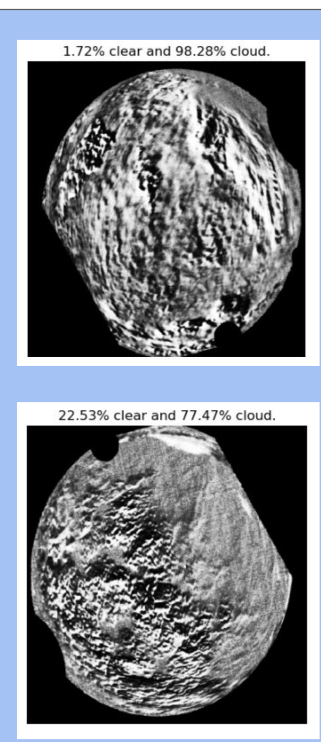
Model training converged very quickly at a high accuracy. The model had a 100% accuracy in identifying clear skies in the test images (0% false positive rate). This makes sense since the true clear sky images are very distinct from those with clouds. It had more difficulty identifying cloudy images correctly but still had 96% accuracy for classifying these in the test images (4% false negative rate). A threshold value of 0.5 was used to differentiate between classifications of clear or cloudy. Example true positive and false negative images are shown below.



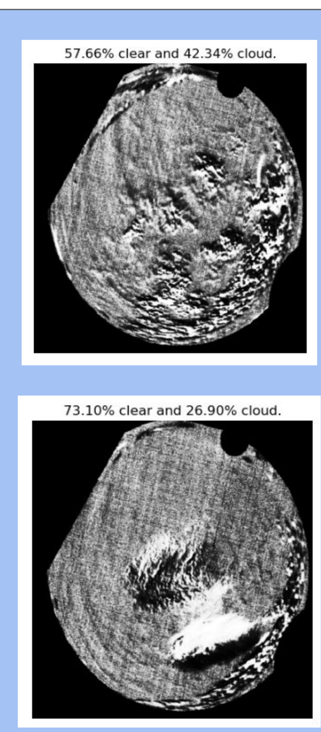
Model Training: Accuracy



Model Training: Loss



True Positives

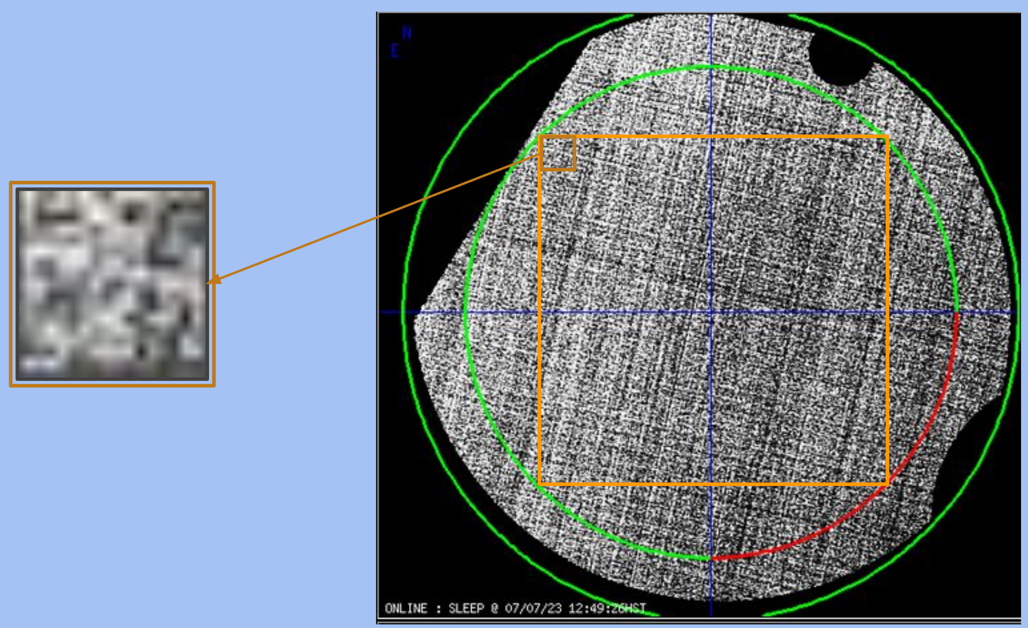


False negatives

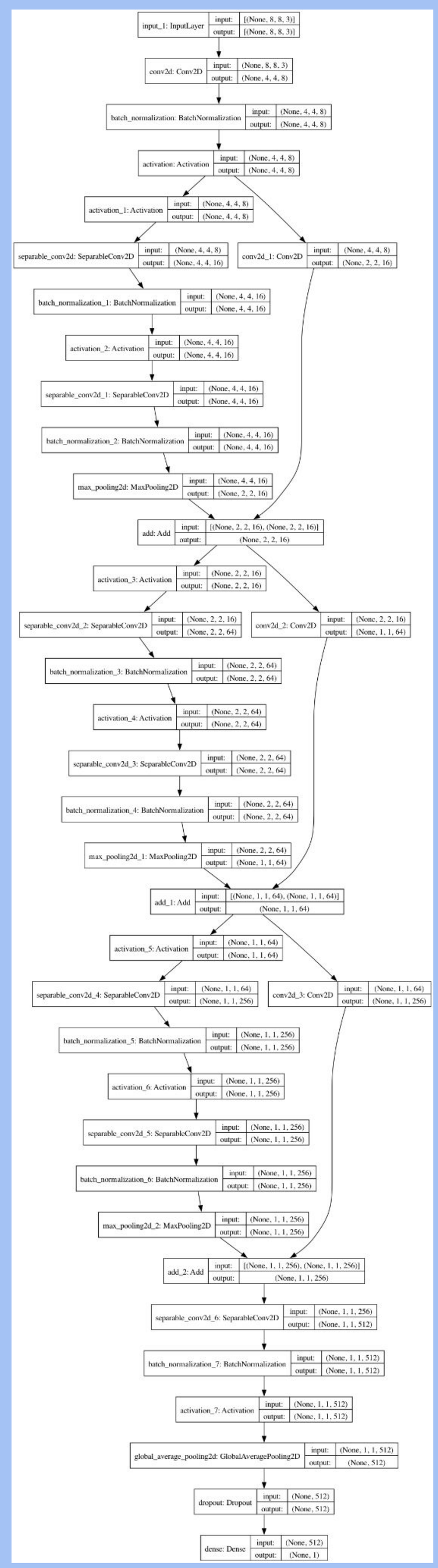
Building a Heatmap Generator

Model and Dataset

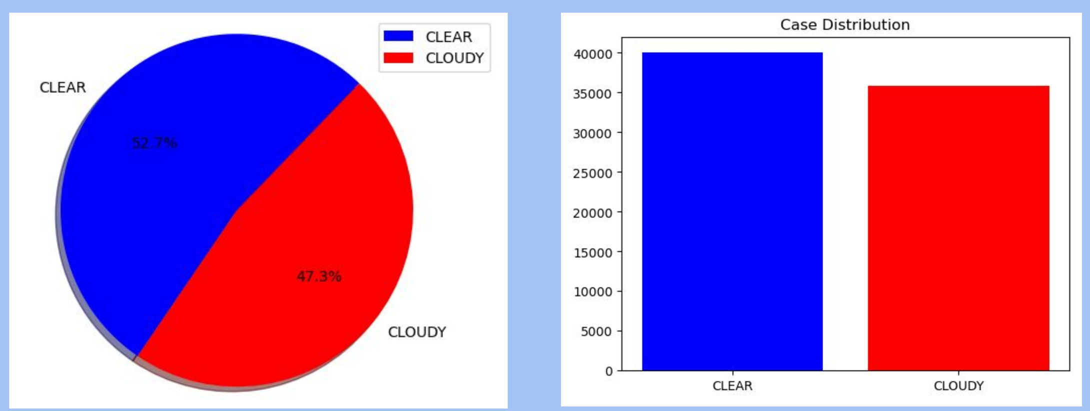
Kernels of sizes 16x16, 12x12, and 8x8 were extracted from various positions within the max squares of images in the dataset of clear and cloudy images. These kernels were classified as clear or cloudy based on the classification of the image they were extracted from. To prevent clear kernels from being extracted from cloudy images and classified as such, cloudy images were chosen that had full or near-full coverage of the sky. Kernels of these sizes were extracted to test the feasibility of receiving a higher resolution output heatmap of a given input image. Model Architecture and Kernel class distribution shown.



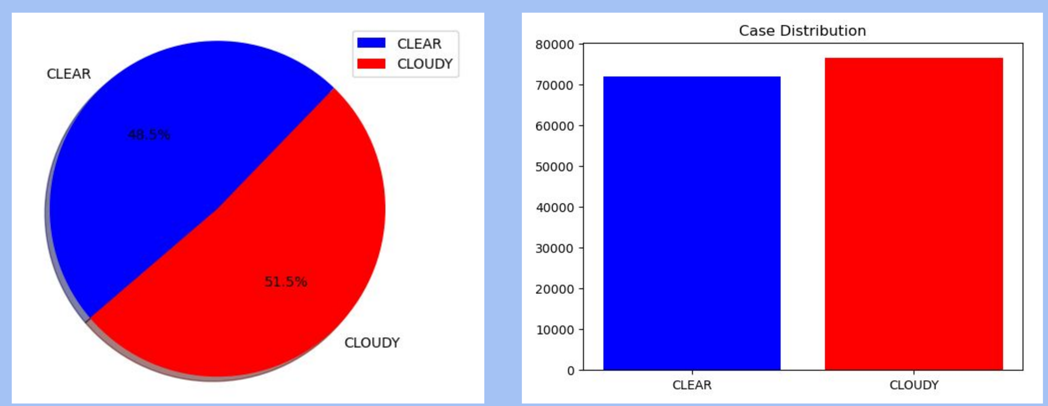
Kernel Extraction from the Max Square of an Image



Model Architecture



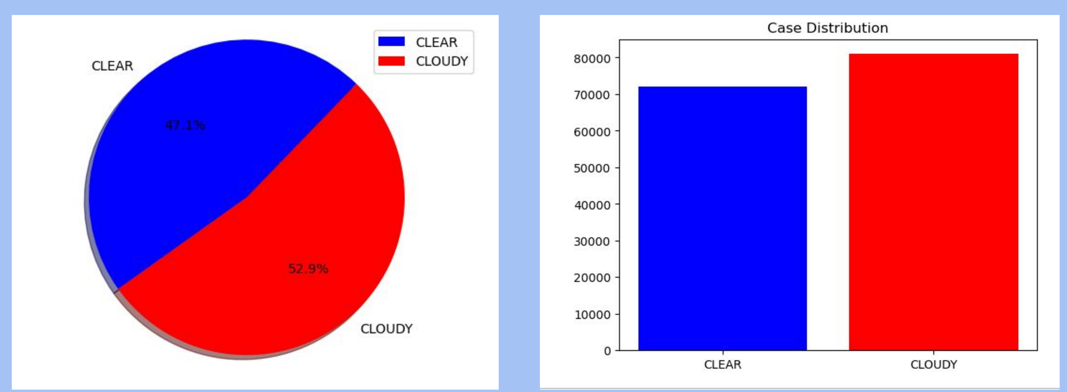
16x16 Kernel Image Class Distribution



12x12 Kernel Image Class Distribution

Initial Results

The first iterations of the heatmap were successfully able to detect clouds in cloudy images. An example input image and the resulting heatmap are shown below. The distribution of prediction values is shown as well as a colored heatmap to better show the distinction between regions reported as clear or cloudy.



8x8 Kernel Image Class Distribution

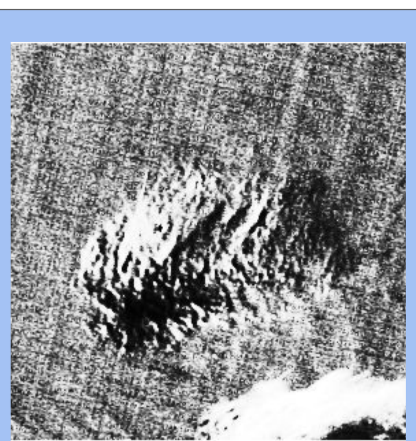
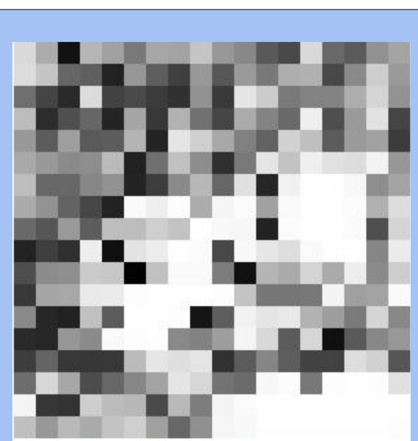
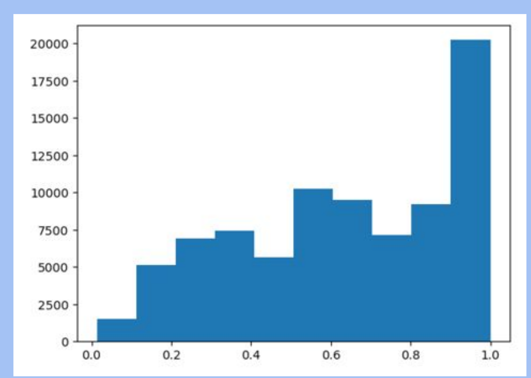


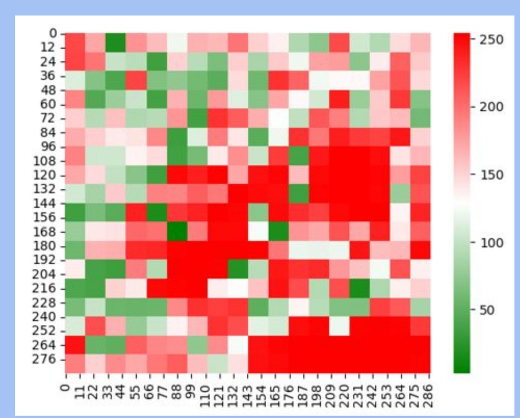
Image Input as Max Square



Heatmap Output



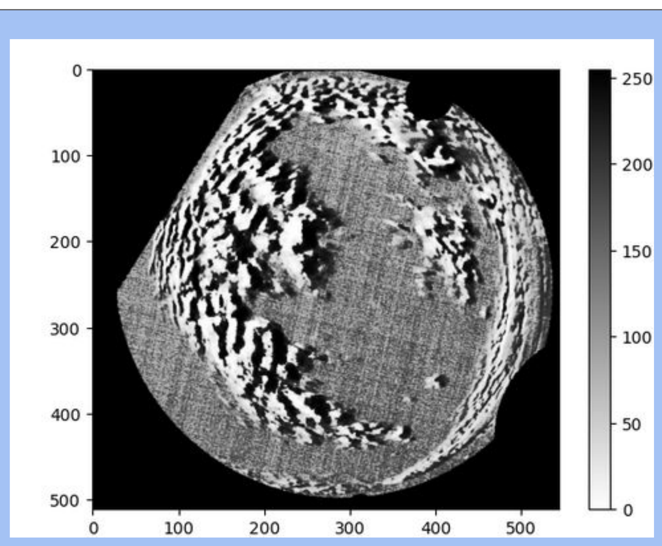
Prediction Value Distribution



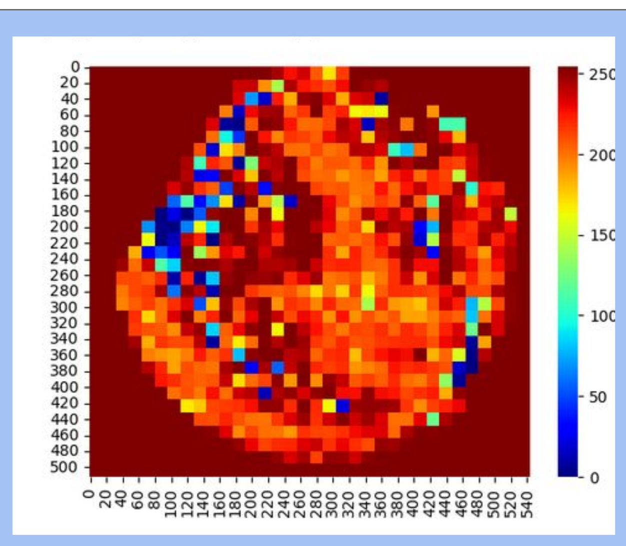
Heatmap Output (Scaled R-G)

Full Image Heatmap

Applying the heatmap program to a full image allows the calculation of a cloud coverage metric defined as the ratio of kernels classified as cloudy divided by the total number of kernels. One issue, however, was the reporting of the masked region of the input as cloudy sections of sky. This was resolved simply enough by excluding the mask in further iterations. Another issue was a skewing of kernels that partially include the mask toward cloudy.



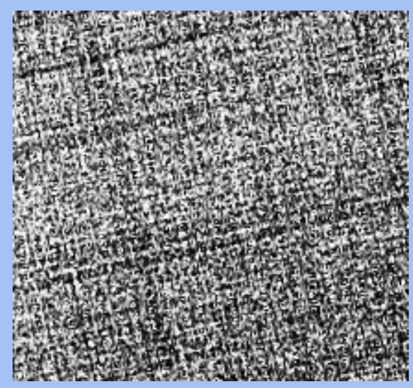
Full Image Input



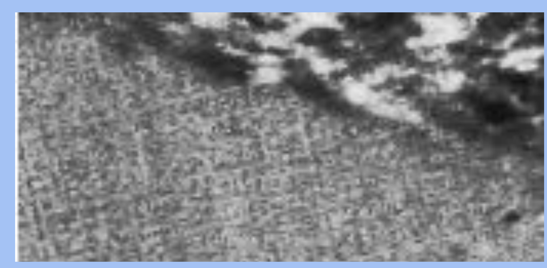
Full Image Heatmap

The Background Problem

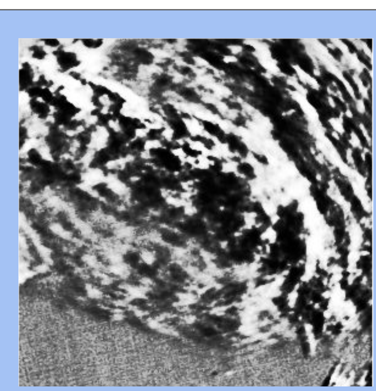
The background sensor pattern is different in clear images vs cloudy images, leading to high prediction values for clear sections of cloudy images. This is shown in the heatmap shown below. The section in the bottom left is clear. However, it returns a value of around 225 out of a scale of 255, indicating a fairly confident classification of cloudy. Initially, this was thought to be due to a histogram normalization process used to create the PNG images used for the analysis from the raw images. However, removing the histogram normalization in the PNG conversion process still resulted in the sensor pattern differentiation. The cause was found to be the limited dynamic range of the FLIR detector changing based on the maximum value recorded. Since cloudy images have higher brightness, the dynamic range shifts up, reducing the range in which it can capture the clear sky.



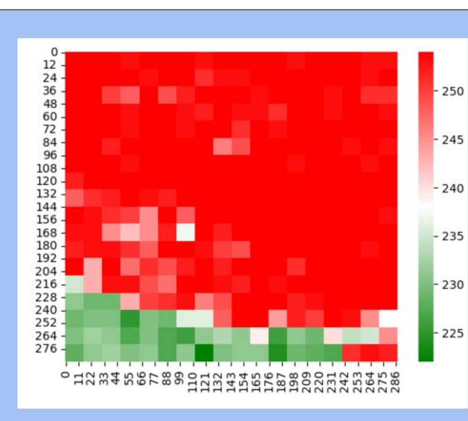
Clear Section of a Clear Image



Clear Section of a Cloudy Image

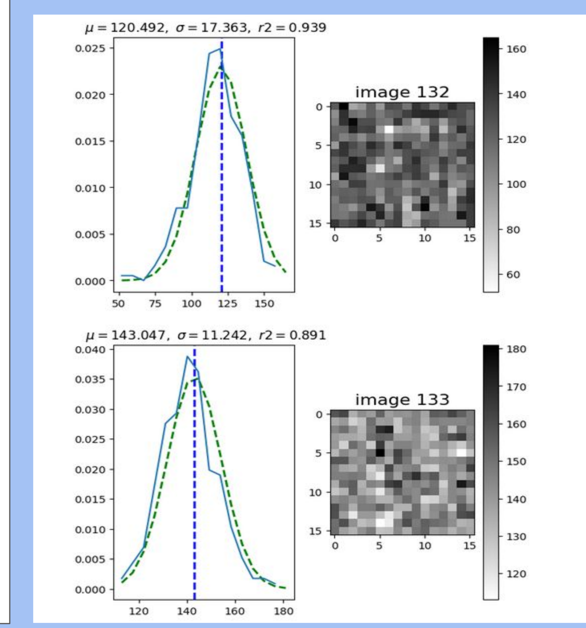


Max Square Input

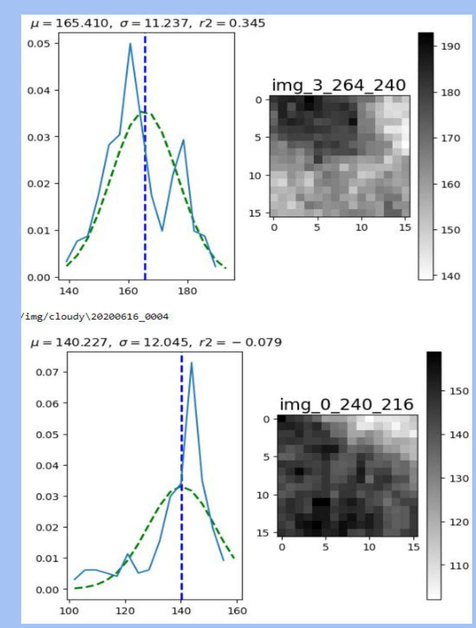


Heatmap of Max Square

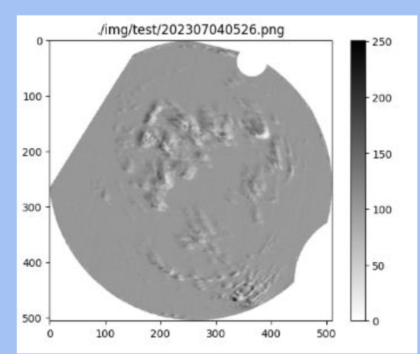
The background issue was addressed based on background information available about the data produced by the ASIVA. Specifically, the fact that clear sky images produce a pixel value distribution to which a Gaussian curve can fit quite well. The difference between 16x16 kernel pixel value distributions of clear and cloudy kernels are shown to the right. By checking the r^2 value of the Gaussian fit for a given kernel's pixel value distribution, we were able to parse through cloudy images to extract clear sections from images that have clouds in them. These images were included in the clear images of the training data set. This resulted in significantly better results as shown below.



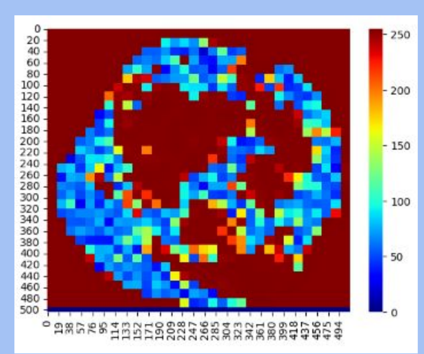
Clear Kernel Pixel Distribution



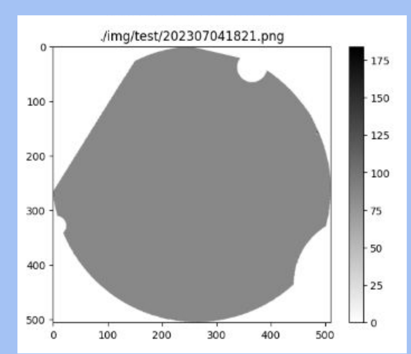
Cloudy Kernel Pixel Distribution



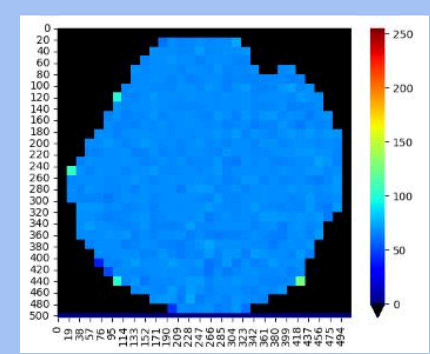
Cloudy Image Raw Input



Cloudy Image Heatmap



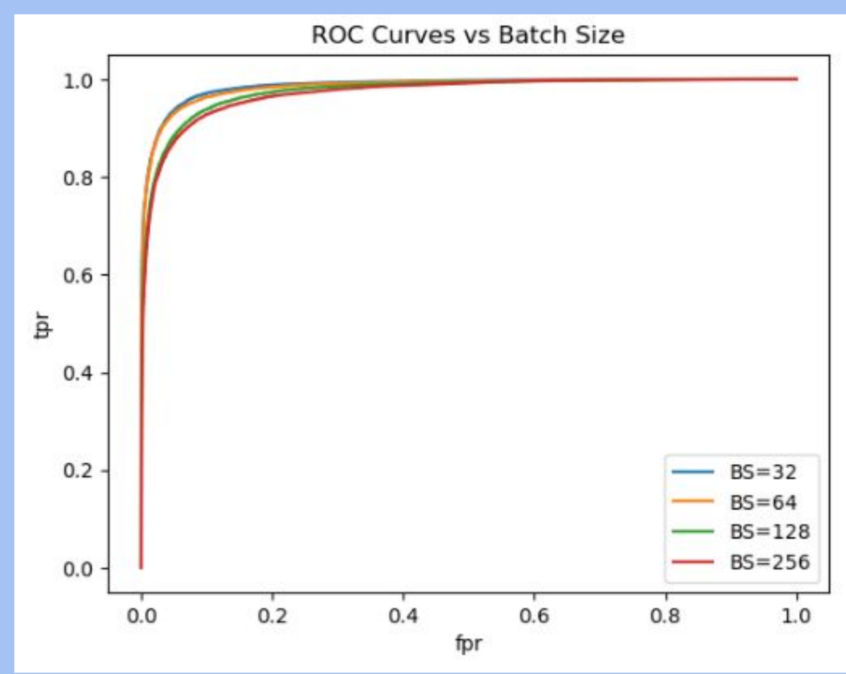
Clear Image Raw Input



Clear Image Heatmap

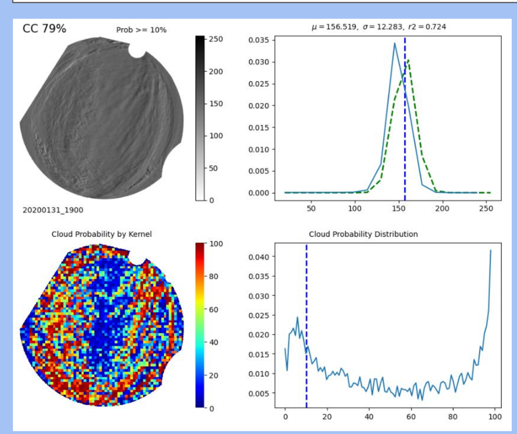
Testing Batch Sizes at 8x8

Different batch sizes for model training were tested using an 8x8 pixel kernel model. 8x8 kernel size was chosen to maximize resolution of the output heatmap. Batch sizes of 32, 64, 128, and 256 were tested. Learning rates were not found to be significantly affected. The ROC curves of each batch size are shown. Performance of each batch size on an example image is shown. Based on the output heatmaps as well as the cloud probability distribution plots, it is apparent that higher batch sizes resulted in a skewing of reported values toward middling numbers.

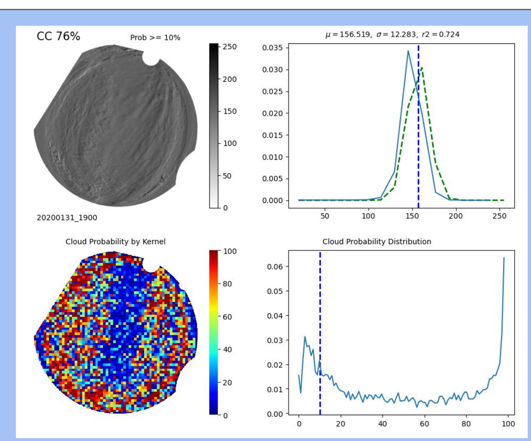


Batch Size	AUC
32	0.987
64	0.984
128	0.976
256	0.972

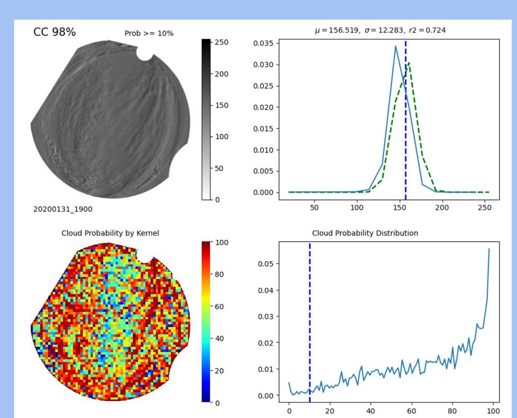
ROC Curves Tested for 8x8 Kernels at Different Batch Sizes



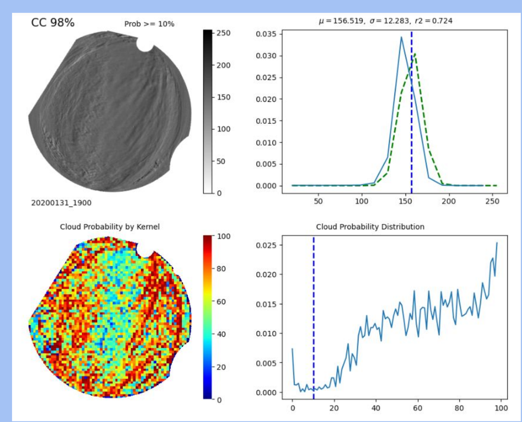
Batch Size = 32



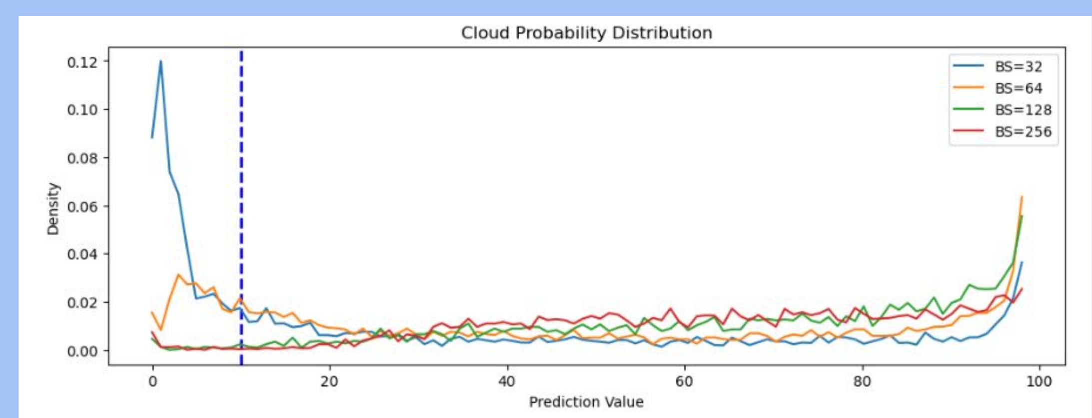
Batch Size = 64



Batch Size = 128



Batch Size = 256



Distribution of Reported Cloud Probability Values for Different Batch Sizes

Conclusion

A batch size of 64 on a kernel size of 8x8 was found to have the best performance. 8x8 pixel kernels maximize resolution while minimizing overlap with the mask. This combination was also found to have the best accuracy in reporting cloud coverage percentage.

References

Abadi et al, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
Brownlee, Jason. "About." MachineLearningMastery.Com, 25 Oct. 2021, machinelearningmastery.com/about/
Model Architecture adapted from: Eda AYDIN, Pneumonia Detection on Chest X-ray Images with Deep Learning (Keras), Nov. 6, 2022, https://github.com/edaaydinea