Night-time cloud classification with machine learning

Akin Kucukkurt

Abstract: Machine learning has been used to detect cloud types and weather forecasts using an image of the sky. This paper focuses on detecting general cloud coverage and cirrus cloud coverage using a decision tree algorithm. An accuracy of 74% is achieved on general clouds and 54% for cirrus clouds and this paper explains these results. This paper provides a simple explanation to help anyone to use and understand the method.

1. Introduction

2. The Data Set

The camera taking the images is the AllSky-340 which has a Kodak KAI-0340 CCD sensor and a Fujinon FE185C046HA-1 lens and it is located on the roof of the East of England Science Learning Centre which is a few hundred meters away from the Bayfordbury telescopes[1]. Information about the images and their classifications are stored inside a csv file.

filename_index	image_number	julian_date	humidity	temperatur	pressure	cirrus	cloud	moon_alt	moon_pha	sun_alt	skymag_r	skymag_v	skymag_b	skymag_c
0	2668441	2457059.451	85	0.9	1023.4	0	0	33	96	-50.24	21.96	21.6	22.25	20.03
1	2668486	2457059.466	86	0.7	1023.4	0	0	35.4	96	-51.88	21.87	21.51	22.2	19.95
2	2668514	2457059.476	87	0.5	1023.4	0	0	36.8	96	-52.71	21.9	21.53	22.18	19.98
3	2668619	2457059.513	89	0	1023.3	0	0	41.1	96	-53.99	21.77	21.47	22.09	19.88
4	2668899	2457059.61	89	0.3	1023.6	0	0	42	96	-44.07	21.48	21.35	21.95	19.68
5	2668941	2457059.624	89	0.3	1023.5	0	0	40.7	96	-41.42	21.61	21.47	22.05	19.78
6	2668945	2457059.626	89	0.3	1023.5	0	0	40.5	96	-41.16	21.62	21.47	22.06	19.79
7	2668983	2457059.64	89	0.1	1023.5	0	0	39	96	-38.45	21.69	21.54	22.11	19.85
8	2668995	2457059.644	89	0.1	1023.5	0	0	38.5	96	-37.62	21.67	21.53	22.11	19.84
9	2669085	2457059.676	89	0.1	1023.7	0	0	33.8	96	-30.77	21.87	21.68	22.24	20.02
10	2669748	2457060.261	80	2.1	1026.3	0	0	-14.4	93	-12.28	22.15	22.05	22.74	20.41
11	2669831	2457060.288	82	1.8	1026.5	0	0	-9	93	-18.27	22.84	22.36	23.48	20.97
12	2669855	2457060.295	82	1.8	1026.5	0	0	-7.6	93	-19.82	22.85	22.37	23.5	20.98
13	2669957	2457060.324	84	1.5	1026.9	0	0	-1.5	93	-26.29	22.89	22.41	23.53	21.01
14	2669995	2457060.336	84	1.4	1027	0	0	1	93	-28.87	22.82	22.36	23.44	20.96
15	2670107	2457060.375	86	1.1	1027.2	0	0	9.3	93	-37.08	22.64	22.13	23	20.74
16	2670124	2457060.382	86	1.1	1027.2	0	0	10.9	93	-38.6	22.6	22.1	22.94	20.69
17	2670167	2457060.4	87	0.9	1027.3	0	0	14.5	93	-41.92	22.53	22.04	22.85	20.62
18	2670187	2457060.408	87	0.8	1027.2	0	0	16.2	92	-43.43	22.49	22	22.8	20.58
19	2670331	2457060.461	89	0.8	1027.6	0	0	26.5	92	-51.1	22.29	21.86	22.6	20.39
20	2670478	2457060.515	90	0.6	1027.8	0	0	34.6	92	-53.67	22.24	21.85	22.5	20.33
21	2670552	2457060.542	90	0.4	1027.9	0	0	37.4	92	-52.56	22.15	21.79	22.46	20.25
22	2670637	2457060.574	91	0.6	1028.2	0	0	39.5	92	-49.35	22	21.73	22.37	20.17
23	2670647	2457060.577	91	0.5	1028.2	0	0	39.6	92	-48.89	21.97	21.72	22.37	20.16
24	2670767	2457060.623	92	-0.3	1028	0	0	39.2	91	-41.43	22.04	21.77	22.41	20.18
25	2670797	2457060.635	92	-0.4	1027.8	0	0	38.6	91	-39.17	22.05	21.8	22.41	20.19
26	2671924	2457061.404	92	3.5	1031.6	3	0	6.2	87	-42.39	22.17	21.91	23.22	20.43
27	2672064	2457061.458	93	2.4	1031.6	0	0	17.3	86	-50.5	22.35	21.99	22.94	20.54

Figure 1: A screen cap of the data-set.

The 'filename_index' stores the name/number of each image,the 'cloud' column has a value from 0-8 representing the classification of general cloud coverage as a fraction of 8 and the 'cirrus' column is the same except for it being only the cirrus cloud coverage. From the 'filename_index' column, 0-4500 are images that are all rated 0 for the 'cloud column' and rated properly in the 'cirrus' column. The general pattern is that about half of this range is rated 0 in the 'cirrus' column with packets of classifications above 0 distributed randomly. The range from 4500-13000 are images that are all rated 0 in the 'cirrus' column and rated properly in the 'cloud' column. The pattern here is about an even number of images for each class with the 8 class having about twice

[1] David Campbell. *Widefield Imaging at Bayfordbury Observatory.* 2010. URL: `http://www-cs-faculty.stanford.edu/~uno/abcde.html`.

as many images. As such 0-4500 can be referred to as the Cirrus only subset and 4500-13000 can be referred to as the Cloud only subset.

3. The Method

For my method I used python with the 'Pandas'[2] module used for easy image and table manipulation, the 'Scikit'[3] for the machine learning algorithm and the 'Squircle'[4] module to turn a circular image into a square. I have split my method into four separate python files,'SquircleApplied.py' which deals with image formating, 'Samples and Features.py' which formats and stores the pixel value into an array to be used by 'Scikit', 'Machine Algorithm Trainer(modified).py' which trains the algorithm using a specific image range and 'Predictor.py' which predicts classifications,compares them and outputs the results in a csv file.

3.1. Cutting and Transforming the image

The initial image from the sky camera looks like this.

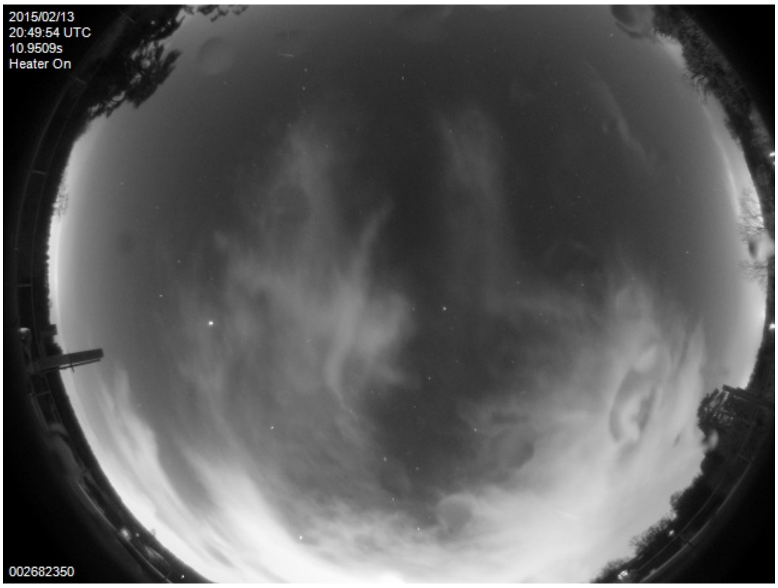


Figure 2: Initial image taken by the Bayfordbury All Sky camera.

It is important to crop this image to remove features that are not necessary for classification as they might interfere with the algorithm and it will lower computation time for this whole method which is already going to be a few hours due to the number of images I'm going to be using. The land features around the edge fit this description. There is also light pollution around the edges which may look like a cloud to the algorithm. The telescope domes at Bayfordbury cannot depress enough to view the sky towards the outer part of the images anyway so it can be cropped out. I then apply a circular mask to crop out the unwanted features and then crop ~~rectangular image into a square that is still~~ masked as circle.

[2]Wes McKinney et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[3]F. Pedregosa et al. "Scikit-learn: Machine Learning in Python ". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[4]B. Verkhovskiy. *squircle*. 2019. URL: `https://pypi.org/project/squircle/`.

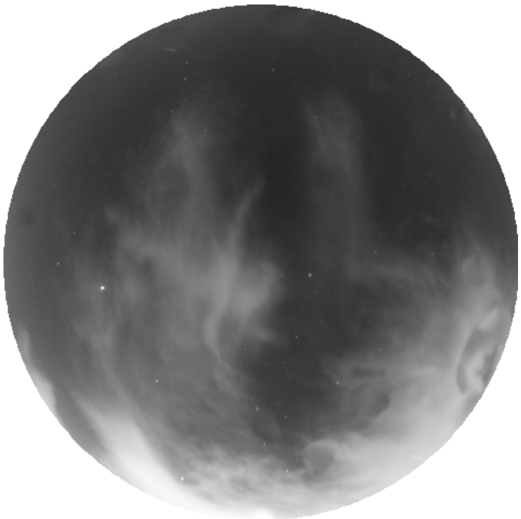
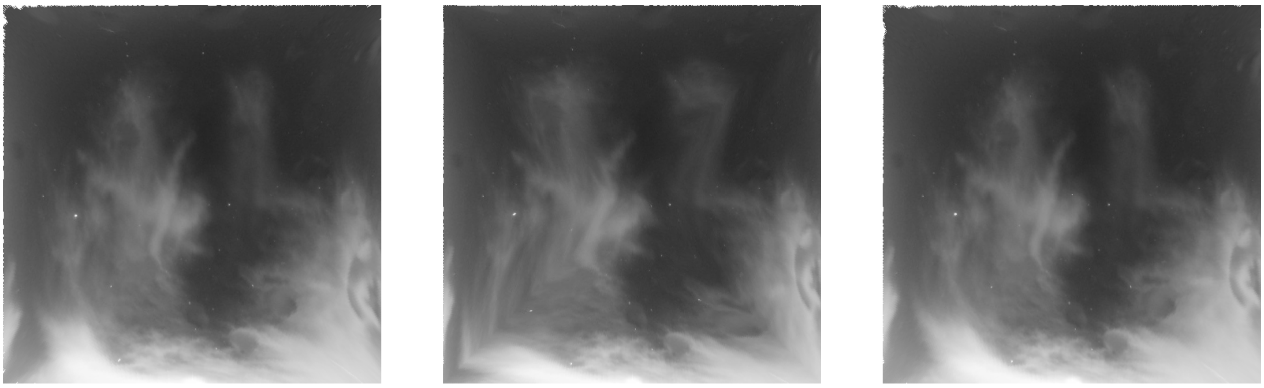


Figure 3: Image with the circular mask applied.

Whilst I could put this straight into the algorithm, The masked area is just empty space and the algorithm's decision trees will be simpler without having to take that and the circular nature of the actual data into account. Ideally I want to turn the actual circular data into a square. In order to do this I use the squircle function. This function only works when the masked image is a square.



(a) FGS (b) Stretch (c) Elliptical

Figure 4: Images for all three 'squircle' methods

The 'FGS' method used is the Fernández-Guasti squircle method which maps the points such that the area is the same and so it has very little deformation. Another method is a simple stretch which doesn't preserve area and so deforms the image.The last method is elliptical grid mapping which is a lot better but I thought that FGS looked more accurate.The differences are minor but you can see one if you look around the middle right most part of the image at the banana-shaped gap in the cloud, while 'FGS' seeems a bit more deformed I thought it was more accurate to the area. The images were classified according to their area in circular form and so area needs to preserved in order for the algorithm to learn properly. There is an obvious issue with all three of these images, namely the white pixels around the edge. There are there because the masked image wasn't a perfect circle since it is made up of square pixels. The last part of my code crops it out and makes the whole process iterable.

3.2. Creating the Learning Array

(How the images were stored in pandas)(Describing the 2 dimensional array wanted by scikit)(How the pixel values were taken out)

3.3. Training the Algorithm

(Mod function)(Slicing and fitting to train algorithm)

3.4. Predicting values

(Mod function)(Slicing and Predicting values)(Analysis of those values)(Moon/other effects)

4. Analysis of Results

The initial and simplest results are as follows.

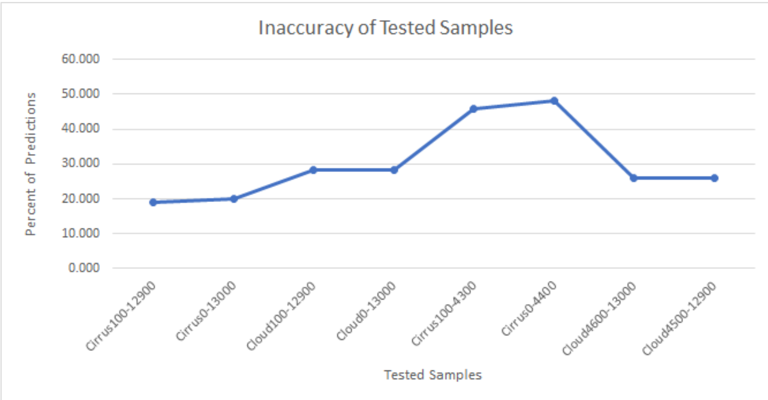


Figure 5: Graph of inaccuracy for each test sample

Test Sample	Inaccuracy Percent
Cirrus100-12900	18.859
Cirrus0-13000	19.923
Cloud100-12900	28.219
Cloud0-13000	28.277
Cirrus100-4300	45.810
Cirrus0-4400	48.272
Cloud4600-13000	25.833
Cloud4500-12900	25.857

Figure 6: Table of inaccuracy for each test sample

The test samples describe which algorithm was used which respond to the 2 columns for the classified values in the data-set.It also describes the range of images that were tested which are a series of 100 images for every interval of 200. So the images that each one was trained on is the opposite series-e.g., tested on 100-12900 means it was trained on 0-13000. The Inaccuracy Percent column describes the percentage of images the algorithm got wrong. This isn't a complete measure as there is no value for the accuracy of the algorithm per image, this is simply the overall accuracy. It is rather misleading and doesn't tell you much how or what the algorithm has learned. However I can use the information about the different subsets that the algorithm has performed on to make some meaningful predictions from this graph.

The Cirrus algorithm has a low accuracy when only the images that were in the Cirrus subset are used. Using what I know of this subset I can say that there isn't an obvious systematic issue with it and so the error is mostly due to measurement error which presents itself as over-fitting as the algorithm learns the error instead getting the general relation. Over-fitting essentially means the signal to noise ratio is low. The signal is the brightness of a cloud and so the total cloud/brightness coverage which means both subsets, Cloud and Cirrus, are looking for the same signal. The noise is the variation of the images in the subset which should be lower for the cirrus subset as it only has images of one type of cloud. I will analyse how true this actually is later on.

Using what I know of the subset and these results I can conclude that the low signal to noise ratio is due to lower number of images. There is a significant difference between the Cirrus100-4300 and the Cirrus0-4400 results, this could be due the differences in the subset they were trained and tested on as well as the difference in the number of images they were trained on as Cirrus100-4300 is trained on 100 more images. The fact that Cloud4600-13000 and Cloud4500-12900 are around twice as accurate and have around twice as much images

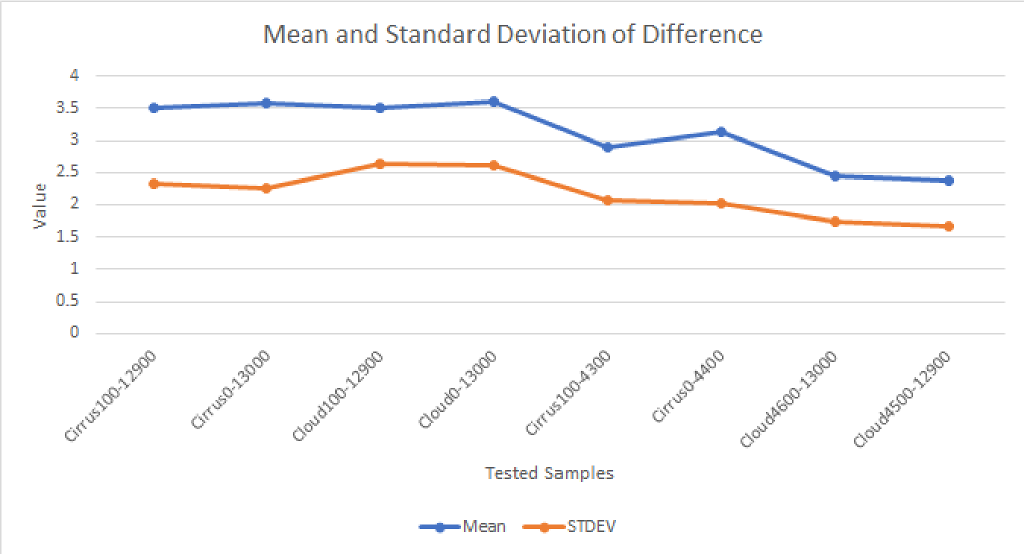


Figure 7: Graph of mean and standard deviation of difference

suggest that the number of images is the dominant factor with the difference in the number of images to accuracy ratio due to the lower variation in the Cirrus subset which decreases noise as well as the differences between the 0-4400 subset and the 100-4300 subset which increases noise.

Although Cirrus100-12900 and Cirrus0-13000 have the best results , this is misleading. The Cirrus subset has mostly zeros and in the Cirrus column the Cloud subset has zero for every image which means that these two algorithms have been trained to label zero so they don't have any practical use.

For the Cloud algorithms those two that were trained only on the Cloud subset have the best result. No doubt that adding the Cirrus subset confused the algorithm as they are all rated zero for Cloud result column. Though the difference between the two is small considering that Cloud0-13000 and Cloud100-12900 have 50% more images that are all labelled wrongly as zero. Around half the Cirrus subset is classified as zero anyway so this helps mitigate the effect of adding the incorrectly labelled subset.

4.1. Analysis of difference between prediction and classification

A simple representation of the difference data between the algorithm and the classification looks like this which is a meaningful representation of how much each algorithm has learnt. These values were calculated by looking at the all wrong predictions and finding the difference between the prediction and the actual classified value for each. Both the Cloud algorithm and the Cirrus algorithm had around the same mean when performing over the whole data-set though the Cirrus algorithm has a slightly lower standard deviation which means it was a bit more consistent but not by much. Even though the Cirrus algorithm has the most initial accuracy we can see from here the effect of the issue that I already mentioned , it's good at classifying zeros but when the image has a value above zero it struggles the most with giving a value near the classified value. The fact that both the Cloud algorithm and the Cirrus algorithm have the same mean and a similar standard deviation is interesting as it would suggest that they both performed similarly even though they dealt with significantly different classifications and data. When I look at a more detailed analysis of the difference later on, there is a significant difference visible between them.

Cirrus 100-4300 and Cirrus0-4400 actually performs better even though it has the lowest initial accuracy. It suffers from over-fitting which means that it has a hard time getting the exact value but the algorithm seems to have learned better as it gets closer to the real value. This makes sense as it performed with properly classified values instead of just zeros. There is slight difference between them which has discussed above I concluded to be from the fact that Cirrus100-4300 was trained on 100 more images which makes it a bit better.

Cloud 4600-13000 and Cloud 4500-12900 has the best results. It has the lowest mean but the standard deviation is slight closer to the mean so it was a bit less consistent which might be due to the increased variation in the Cloud subset. The low mean is consistent with the fact that it performs on a properly classified subset and has the most number of images. The difference between these two and the Cirrus only algorithms before it isn't as much as the difference in initial accuracy. Even though the Cloud algorithm is twice as accurate overall it hasn't learnt twice as much which means that the Cirrus subset has a systematic issue that is causing it to be less accurate overall which isn't something I was able to predict in the first part of this section. I will analyse where this comes from later on.

A more detailed representation is this which compared high and low differences between predicted and actual values.

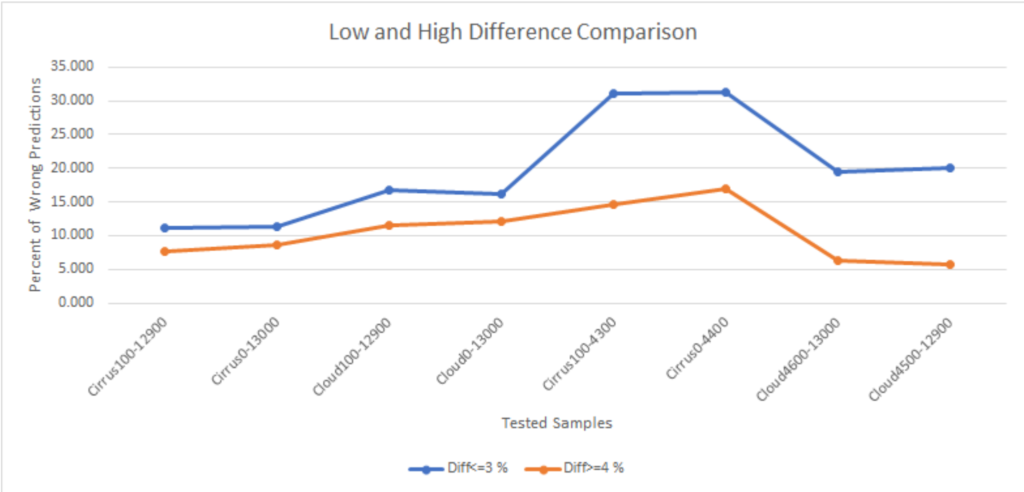


Figure 8: Graph of high and low difference

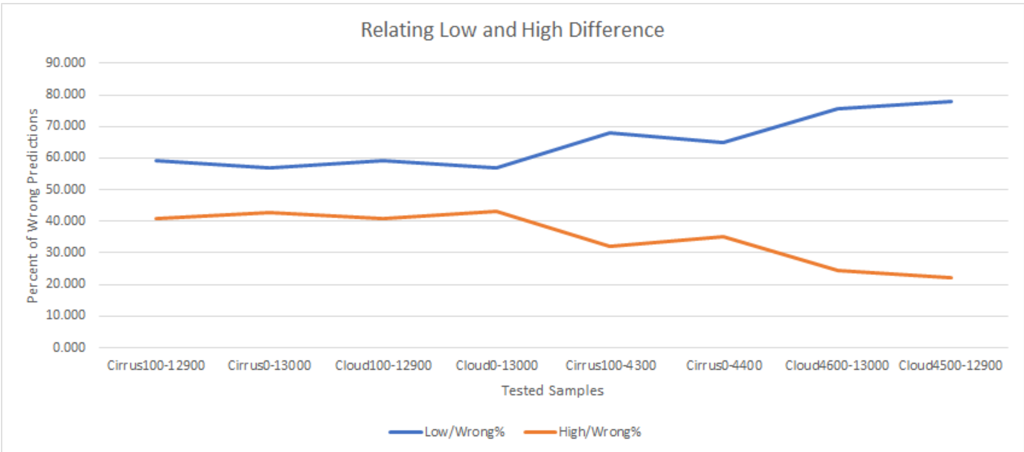


Figure 9: Graph showing what percentage of wrong predictions were high and low

The first graph represents the percentage of predictions that had a difference of 0-3 from the classified as the top line and the bottom represents the predictions that had a difference of 4-8. Over-fitting will cause the algorithm to make small mistakes so the low difference line can be approximated as the size of the over-fit error. The high differences are due to a systematic error as it approximately only occurs when the classification has a large error which suggests a mistake. These two can have some overlap but the boundaries I have chosen are approximately where each is respectively dominant. In the case of the two algorithms working on the entire data-set though the systematic error effects the low difference range as well which means the first four points on this graph is misleading, as it suggest the dominant issue was over-fit when it really wasn't. For that reason I won't be analysing them. An easy way to relate the two lines is to look at what percentage of the total error they make up which leads to the second graph.

The graph helps confirm what I predicted about over-fit being the main issue. Cirrus100-4300 has a bit more over-fit error than Cirrus0-4400 which is good as it confirms the effect of having more images as I predicted in the first part of this section. These two have more systematic error than Cloud4600-13000 and Cloud4500-12900 which confirms what I concluded above that the Cirrus subset has a higher systematic error than the Cloud subset.

I can get some more info by looking at the specifics of the low difference.

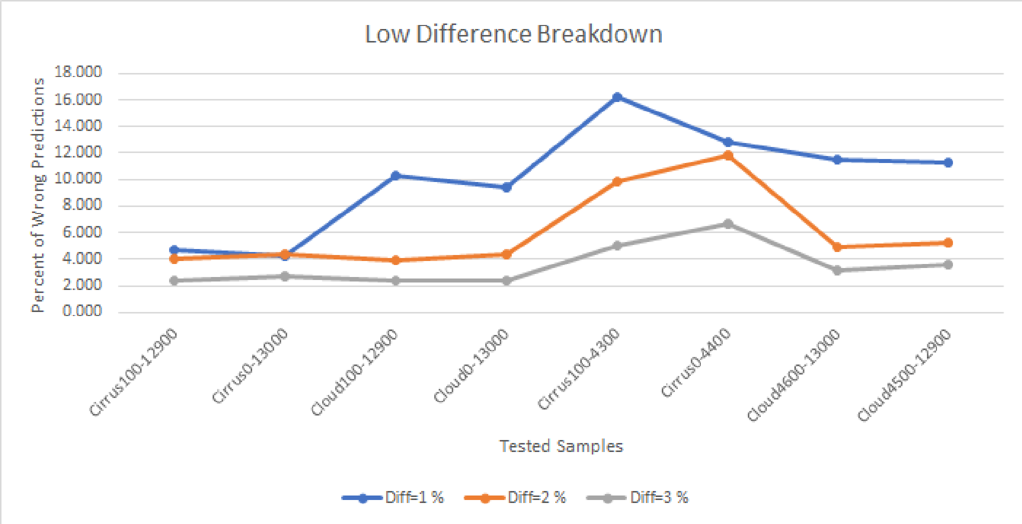


Figure 10: Graph showing differences of 1-3.

Cirrus100-12900 and Cirrus0-13000 stands out with an abnormal distribution due to it's flawed nature and there is finally a difference between these two and Cloud100-12900 and Cloud0-13000 which has a normal looking distribution which shows that it isn't as flawed.

Cirrus100-4300 is significantly better than Cirrus0-4400 which is a repeat result. These 2 results however seem to be inflated in comparison to Cloud4600-13000 and Cloud4500-12900 which again is most likely due to having a lower number of images leading more over-fit which is represented in this graph.

In conclusion the best algorithm seems to be Cloud4500-12900 which has the most number of images and even though it suffer from increased variety , the large amount of images seems to make up for it. Training over the whole data-set when it isn't defined for both parameters predictably leads to failure. I also can't really measure the error from David's classification from these results as it's effects essentially blends in. The Cirrus subset seems to suffer from increased systematic error well as over-fit error.

4.2. Image Analysis

I can find an explanation for the systematic error by analysing the image sets. First of all though, it's important to mention to algorithm had no problem dealing with the moon, when looking through the list of images that the algorithm got wrong, virtually none of them were of the moon.

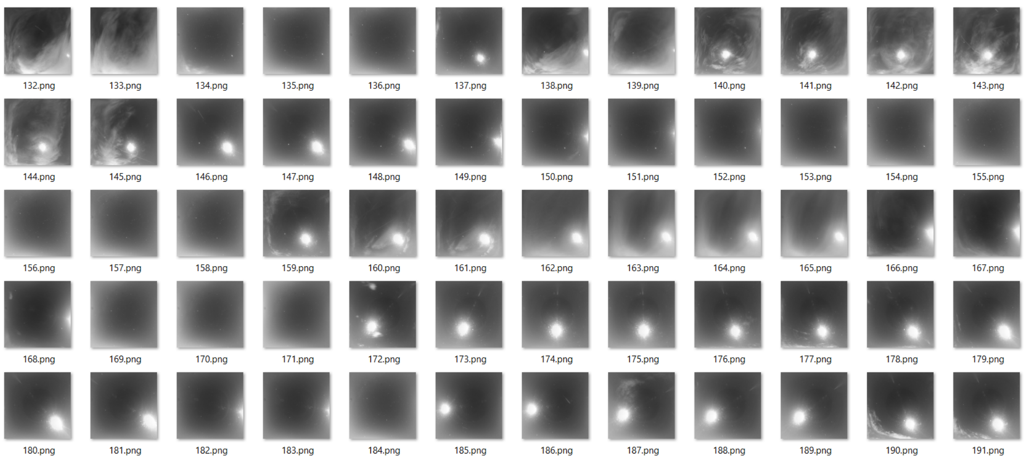


Figure 11: Screen cap showing images with a moon inside them.

The explanation for this is that the moon is much brighter than any cloud so the algorithms learn to ignore it.

The algorithms, especially the Cirrus only algorithm can't tell the difference between light pollution and clouds.

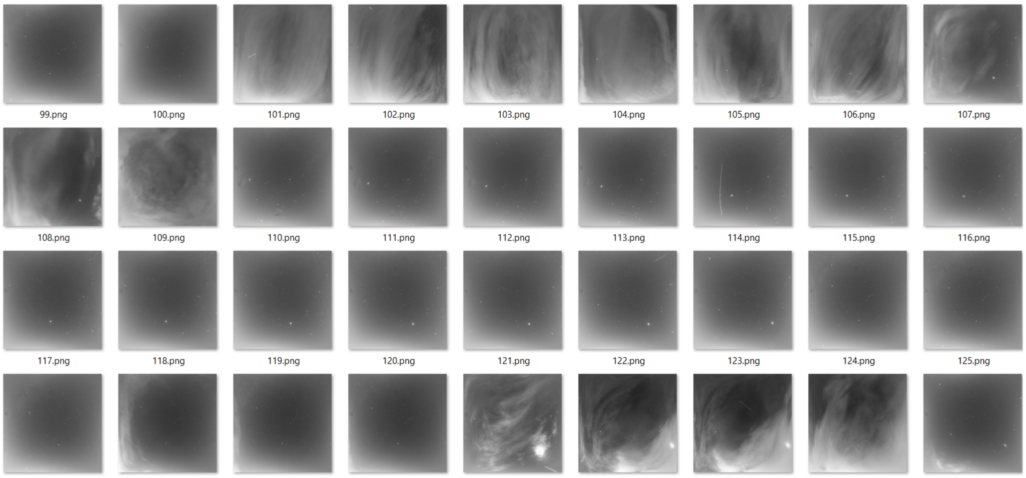


Figure 12: Screen cap of some images showing light pollution.

The information the algorithm has is the gamma value of each pixel and light pollution has the same effect as clouds, Cirrus especially struggles because they tend to have low brightness and are more likely to send the same signal as light pollution. The algorithm did properly predict some of these images but it was very inconsistent.

There is another problem with the Cirrus set, the algorithm struggled to identify thin cirrus clouds.

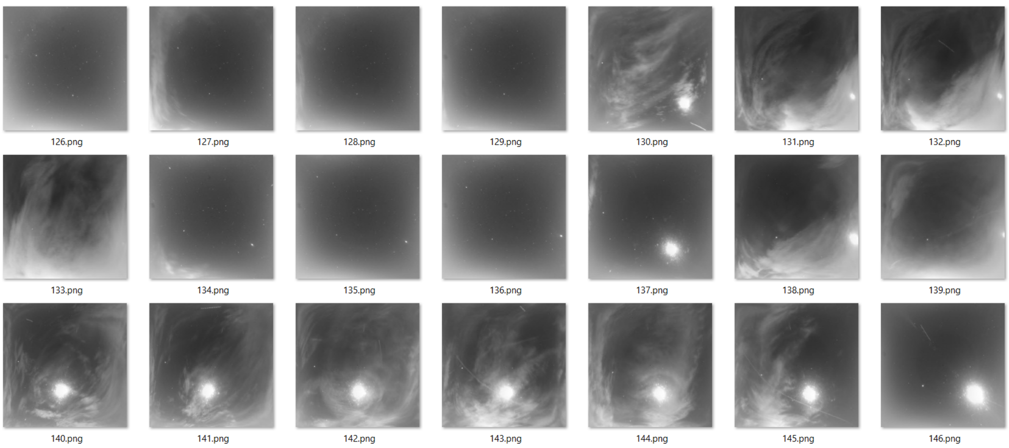


Figure 13: Screen cap of some images shwoing some thin cirrus clouds.

Including thick Cirrus clouds like 101.png and David labelled very light clouds as zero which seems to have confused algorithm as it's essentially getting two separate signals. Couple that with the issue discussed before and the fact that it has less images, I can see why the Cirrus algorithm had a large systematic error and the lowest overall accuracy. The entire reason of having a cirrus category was to identify a type of cloud that doesn't interfere much with telescopic observations but thick cirrus clouds do have a large interference so they should actually not be included in this category

Earlier I predicted that the Cloud subset should have more variety and therefore suffers from it's effects. It has any type of cloud inside it, including cirrus.

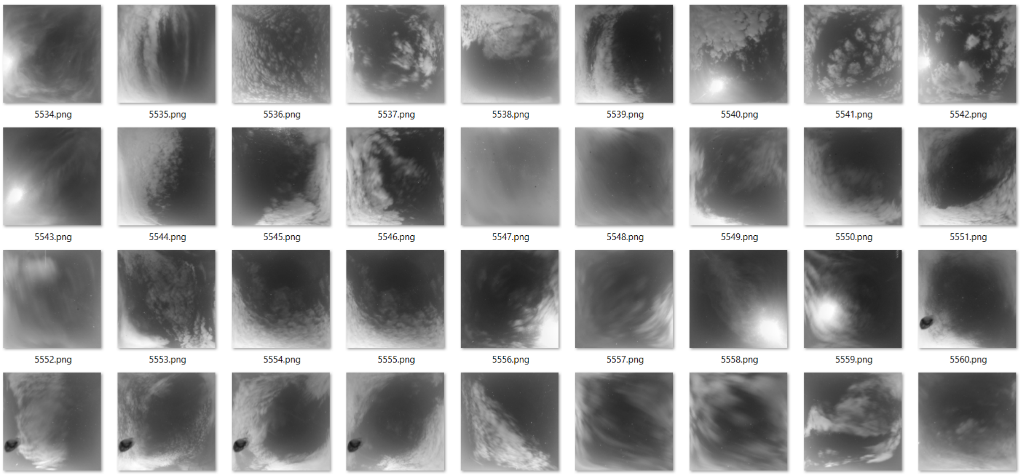


Figure 14: Screen cap of images showing cirrus clouds.

Every Cloud image is classified as zero in the Cirrus column and with this type of variety that kinda looks likes the Cirrus set, I can put another reason on why Cirrus0-13000 and Cirrus100-12900 was a failure. However the images of cirrus clouds in this subset are fairly thick and bright, despite the variety in cloud types and shapes, the Cloud subset has clouds that are of similar brightness which means this subset doesn't suffer from the issue mentioned above despite seemingly having more variety. Also whilst about half of the Cirrus set is classified as zero, the cloud subset is about equal numbers of images for each classification with the '8' class being about twice as big as the rest. You can see more evidence for these reasons with this image.

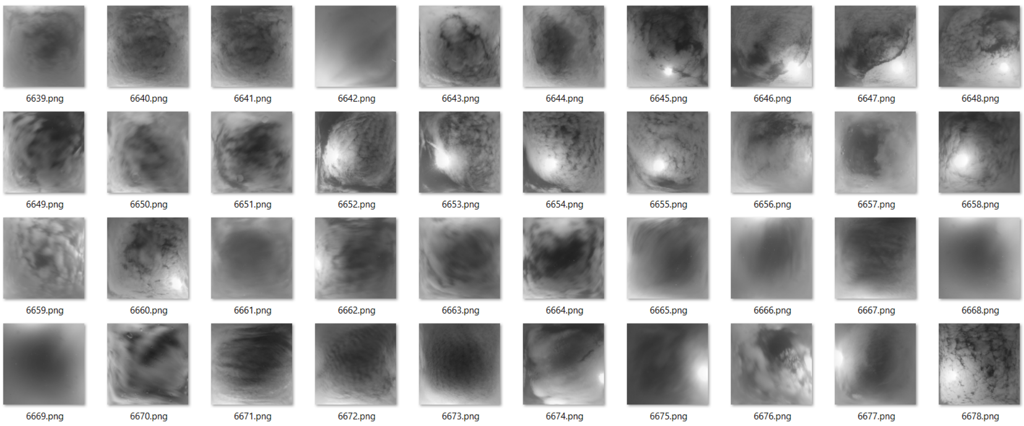


Figure 15: Screen cap of images showing varied cloud cover.

Whilst I can now tell why the Cirrus subset was more flawed than the Cloud subset, this image analysis allows me to see what was wrong with the classification system overall and how I can improve the algorithms. The classification system needs to both become more simple. Having a specific measure in 1/8's for cloud coverage complicates the signal and introduces more measurement error. Most images won't have an exact multiple of this coverage and classification by human eye introduces a large error, the best way to reduce this is to simplify the classification system by reducing the number of classes. For example doing 1/4's or even 1/3's and the aim of this algorithm is to tell whether or not the sky is cloudy for telescopic observations which you can still do as you can define new boundaries. The issues with the Cirrus subset can be resolved by only including thin low brightness cirrus clouds as well including more images that don't have the classification of '0'. Of course increasing the overall number of images will also improve the algorithm and whilst there is a point of diminishing returns, I don't believe this data-set reached that point. Finally I could have used other methods like 'Random Forest' which are similar to decision trees but seem to be better[5], one paper uses this

[5] Houtao Deng. *Why random forests outperform decision trees.* 2018. URL: https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5.

method and 717 samples to predict cloud types and achieves a good result of around 70%[6]. Another paper has an exploration of different methods for predicting cloud coverage including random forests[7].

4.3. New Results

To apply some of the improvements mentioned above, the first step I took was to simplify the classification system. I tested two types of classes, type-2 in which the classifications are reduced to 0 and 1 where 0 is equal to the previous class of 0 and 1 is equal to the other previous classes (1,2,3,4,5,6,7,8) and type-3 in which the classifications are reduced to 0,1 and 2 where 0 is equal to the previous class of 0, 1 is equal to previous classes 1,2 and 2 is equal to the rest of the previous classes (3,4,5,6,7,8).The type-2 classification isn't practical for the aim of this code as it just detects any presence of cloud whereas the aim is to be able to distinguish between no cloud where the sky is suitable for observations, small amounts of cloud where the sky is still suitable for some observations and large amounts of cloud where the sky is not observable. This system is represented in the type-3 classification so it is what I'll be using from now on.

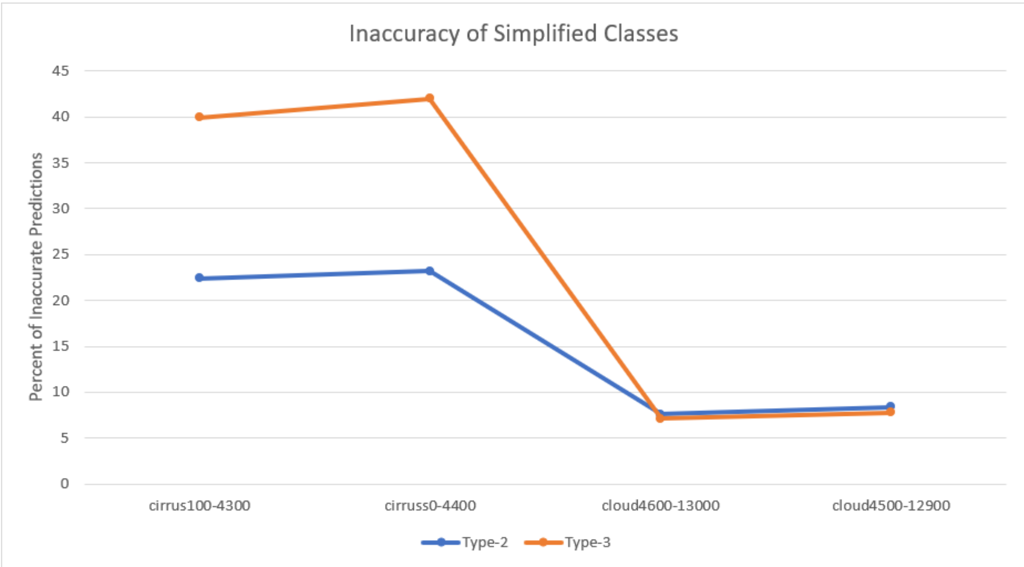


Figure 16: Graph showing the inaccuracy of the new class types.

There is a vastly lower inaccuracy for the simplest system, type-2, for the cirrus subset. Type-2 is a system which reduces over-fit error as it boosts the signal in the signal to noise ratio which means that this vast difference indicates that the cirrus subset suffers from large amounts of over-fit error. In comparison to the original classification system there is an improvement of about 5-6% (for type-3) for the cirrus subset which is a much smaller difference that is accounted for by the fact that the 3,4,5,6,7,8 classes are very small in size in the cirrus subset which means that type-3 doesn't boost the signal as much, the addition of the 1,2,3,4,5,6,7,8 classes in type-2 is much bigger so it has a larger effect. As for the cloud subset there is an improvement of 18% (for type-3) when compared to the original classification system which tells us that a significant portion of the error was due to over-fit which is to a lesser extent than in the cirrus subset. Type-3 performs slightly better than type-2 in the cloud subset which indicates that the impractical nature of type-2 leads to a systematic error ,perhaps due to the unevenness of the class sizes, where it can also be inferred that the remaining error in the cloud subset is not due to over-fit as type-2 decreases over-fit error ,more than type-3, but here it has very little effect in comparison to type-3.

The next step was to test out different depth sizes.

[6] "Automatic Cloud-Type Classification Based On the Combined Use of a Sky Camera and a Ceilometer". In: *Journal of Geophysical Research:Atmospheres* 122 (2017).

[7] Lerch S. Ayari M.E. Baran S. Baran Á. *Machine learning for total cloud cover prediction.* 2020. URL: https://arxiv-org.ezproxy.herts.ac.uk/abs/2001.05948.

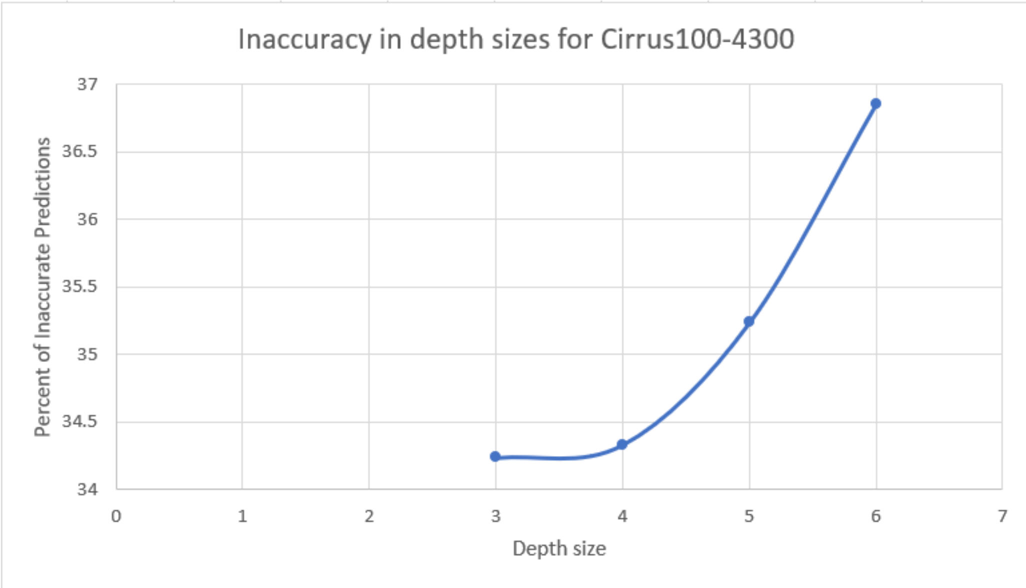


Figure 17: Graph showing the inaccuracy of different depth sizes.

Acknowledgements

I would to specially thank David Campbell setting up the All Sky camera as well as classifying the images used in the data-set. Finally I wish to thank my friend Orkan Mustafa for helping me with some programming issues.

References

"Automatic Cloud-Type Classification Based On the Combined Use of a Sky Camera and a Ceilometer". In: *Journal of Geophysical Research:Atmospheres* 122 (2017).

Baran Á., Lerch S. Ayari M.E. Baran S. *Machine learning for total cloud cover prediction.* 2020. URL: `https://arxiv-org.ezproxy.herts.ac.uk/abs/2001.05948`.

Campbell, David. *Widefield Imaging at Bayfordbury Observatory.* 2010. URL: `http://www-cs-faculty.stanford.edu/~uno/abcde.html`.

Deng, Houtao. *Why random forests outperform decision trees.* 2018. URL: `https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5`.

McKinney, Wes et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference.* Vol. 445. Austin, TX. 2010, pp. 51–56.

Pedregosa, F. et al. "Scikit-learn: Machine Learning in Python ". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

Verkhovskiy, B. *squircle.* 2019. URL: `https://pypi.org/project/squircle/`.

Appendix A.

..