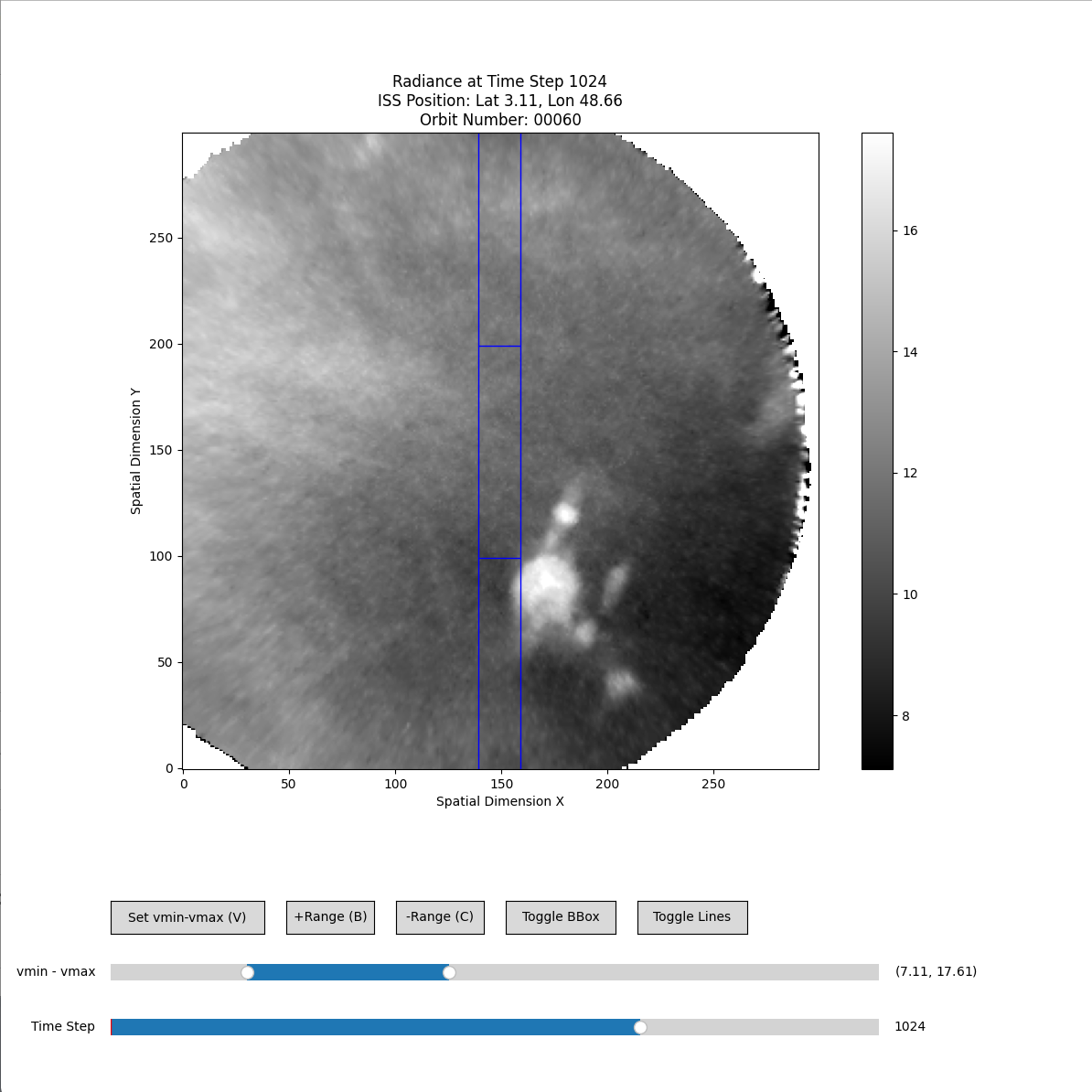
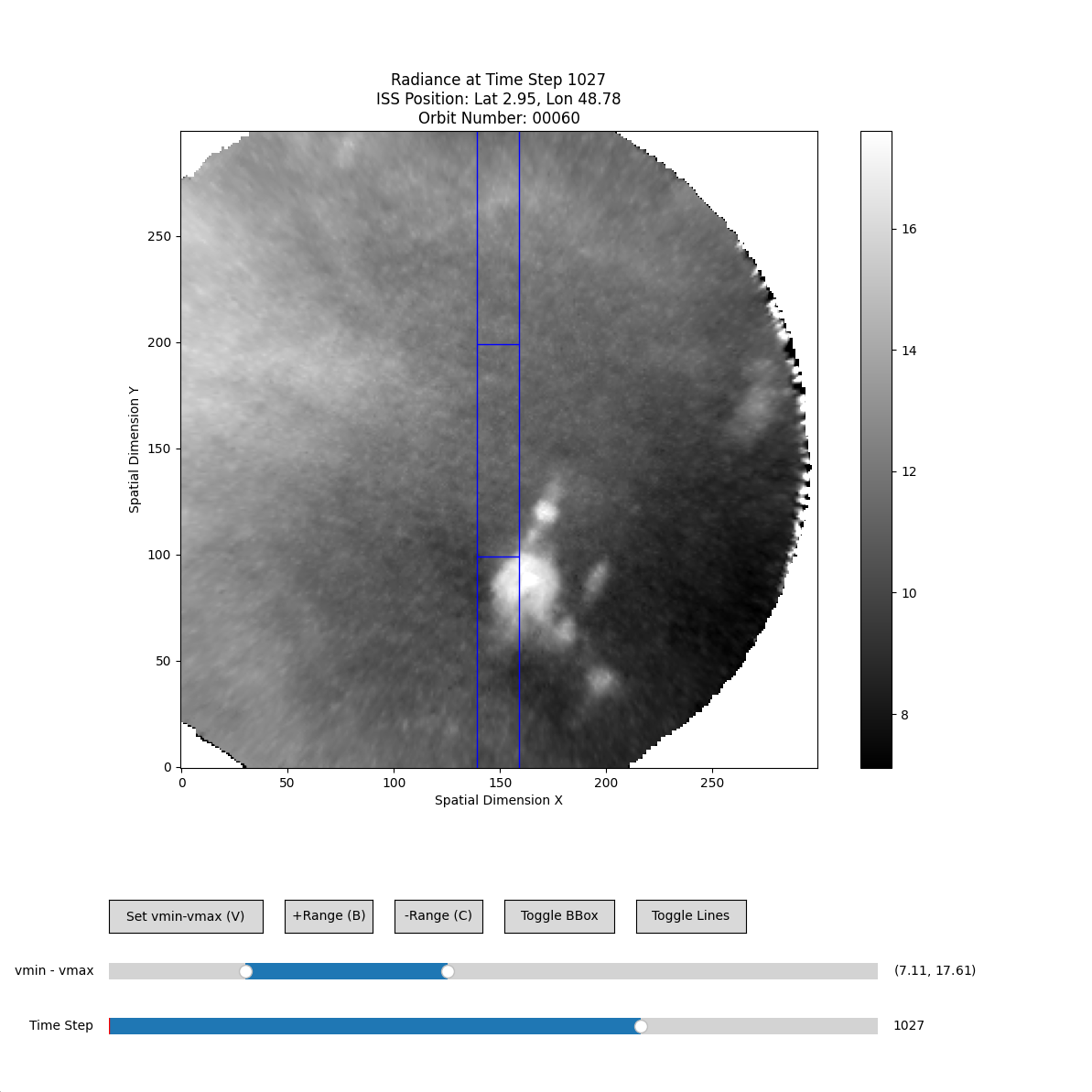
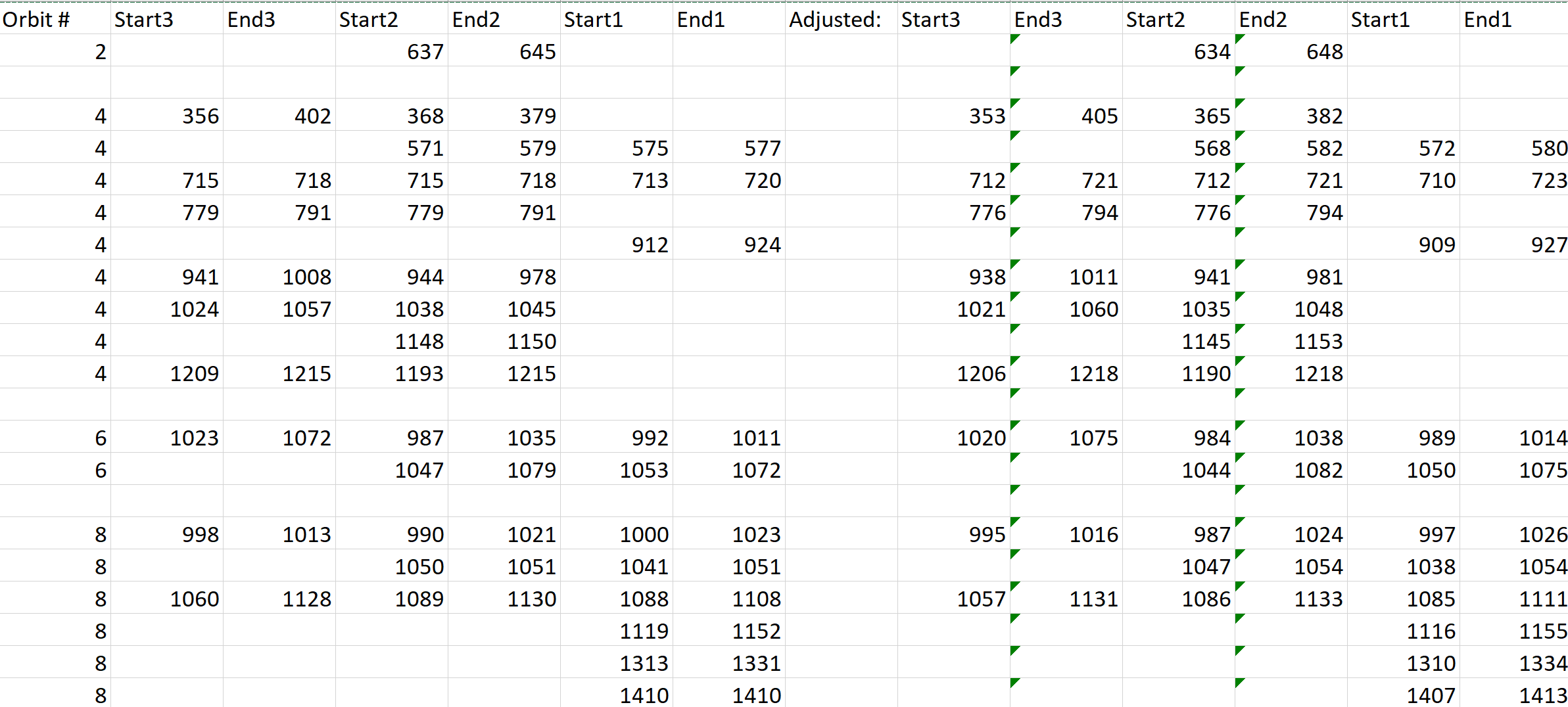
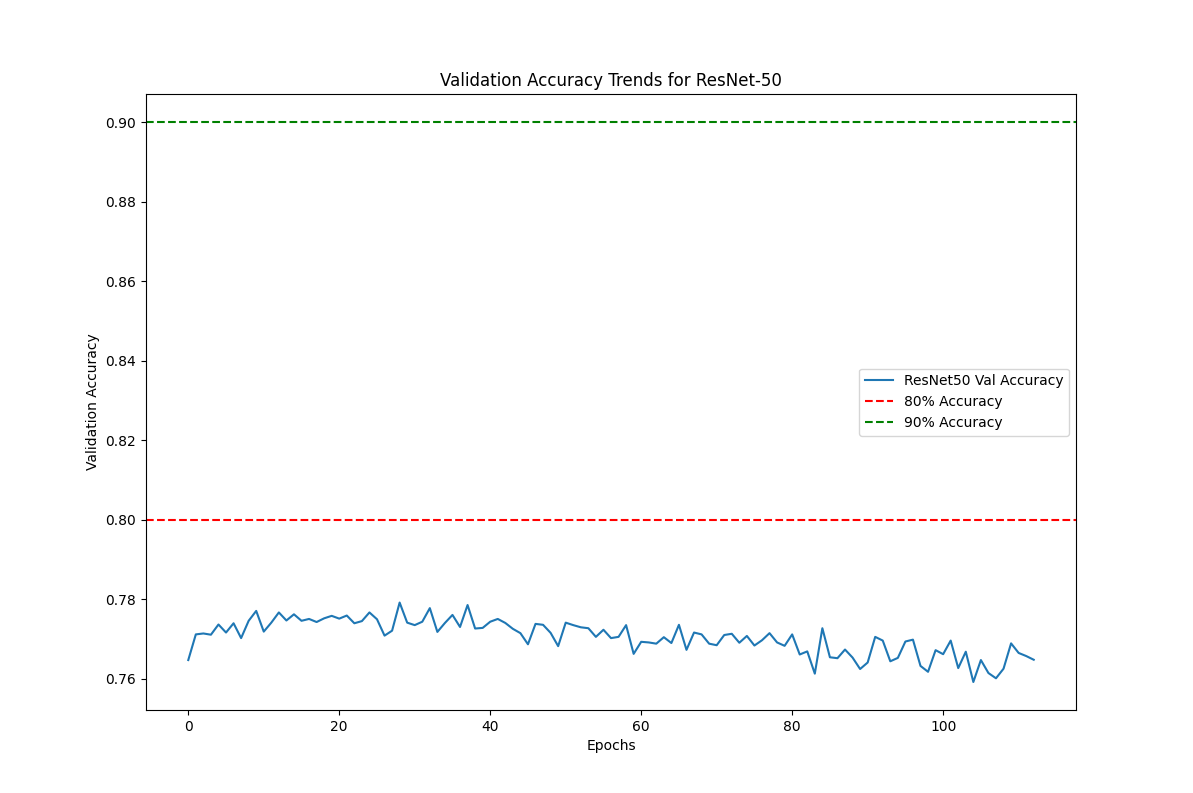
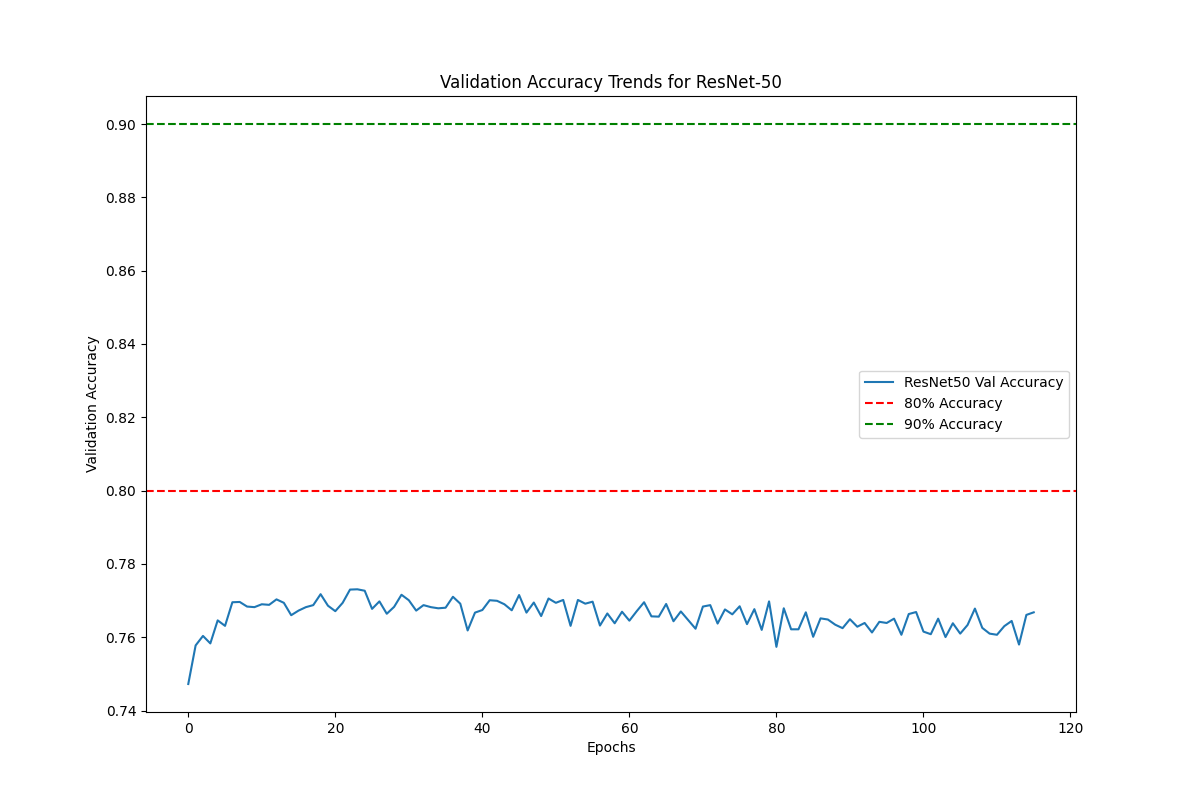
Dallin’s current criterion: clouds are 5 frames into the box. (We carried this criterion from image classification, when ‘cloud’ is when the clouds are 5 frames inside the big image).

Adjust: 5 -> 2 frames.

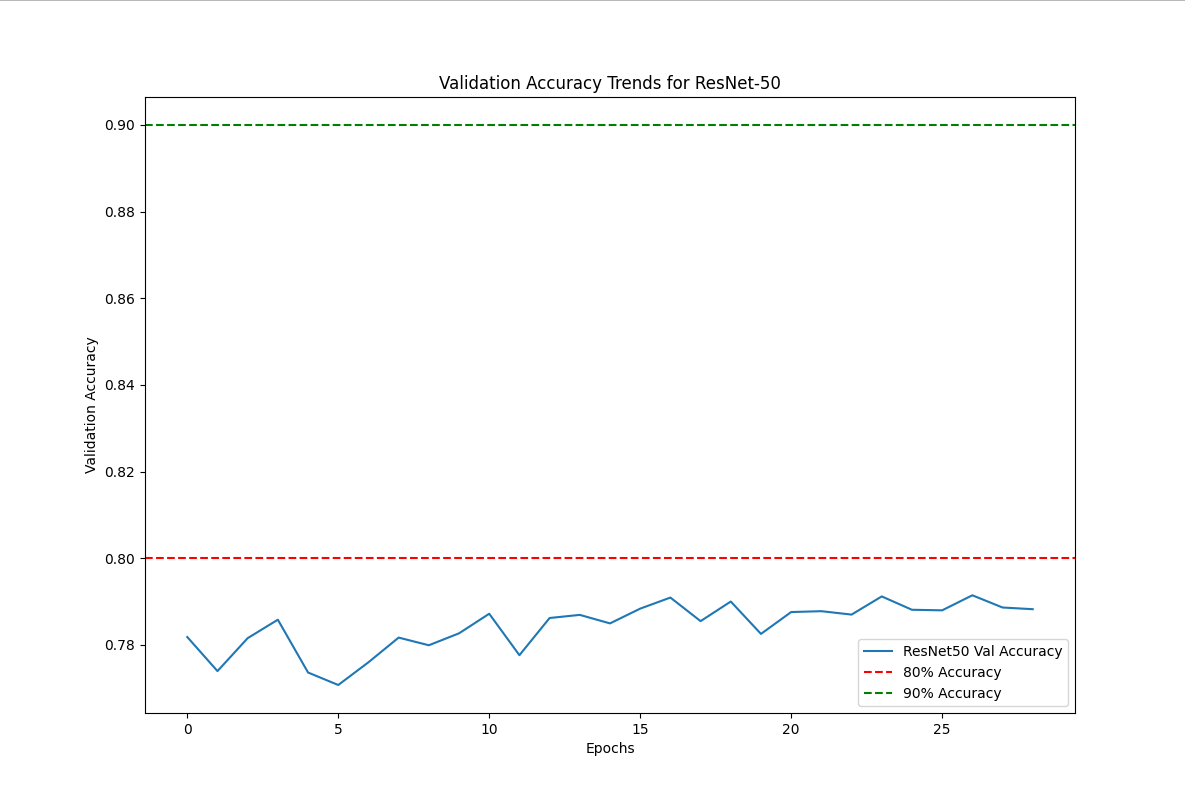








With 75 corrected and 90 added.



The problem of imbalanced dataset.

cloud\_chance=0.25, clear\_chance=0.025

Clear 55,938 Files

Cloud 20,482 Files

Making the decision boundary between ‘cloud’ and ‘clear’ clearer.

Adding a function to check if a frame is clear (more than 5 frames away from cloud intervals)

Accuracy increased (but the test set is also different compared to before)A graph with lines and numbers

Description automatically generated

A graph with a line and a line

Description automatically generated with medium confidence

Time needed to predict all subfolders: 59015.98 seconds

**Uploaded the files to the soc server at Z:\soc\l1c**

Attempt to smooth the curve.

* Previously, when doing only binary prediction, I took running average of 5.
  + If running average > 0.5 then binary prediction is cloud.
* I also applied a filter, so that it’s only cloud if there are at least 5 consecutive frames.
* When switching to integer probability, running average was kept but filtering wasn’t applied.
* Reapply filtering to make the prediction less jumpy.
* Filtering will depend on the threshold value. So, the users will have to implement filtering themselves.

What to do: ‘dynamic’ vmin-vmax (radiance value)

What is currently saved:

A screen shot of a graph

Description automatically generated

What we see when looking for clouds:

A screen shot of a graph

Description automatically generated

vmin\_default = np.percentile(radiance\_flat, 0.4) \* 0.96

vmax\_default = np.percentile(radiance\_flat, 99.7) \* 1.05

* May help increase the accuracy.