Supergranule Segmentation and Analysis

by Andrew Ngo, University of Delaware

Introduction

Supergranulation is a fluid-dynamical phenomenon taking place in the sun's photosphere. Supergranules have been researched to typically have a diameter of 20-30 Mm, dynamical evolution time of 24-48 hours, and a strong 300-400 m/s horizontal flow but weaker 20-30 m/s vertical component.

Modern developments of supercomputing resources and observatories providing high spatial and temporal resolutions of solar surface dynamics have greatly improved conditions for making progress with supergranulation research.

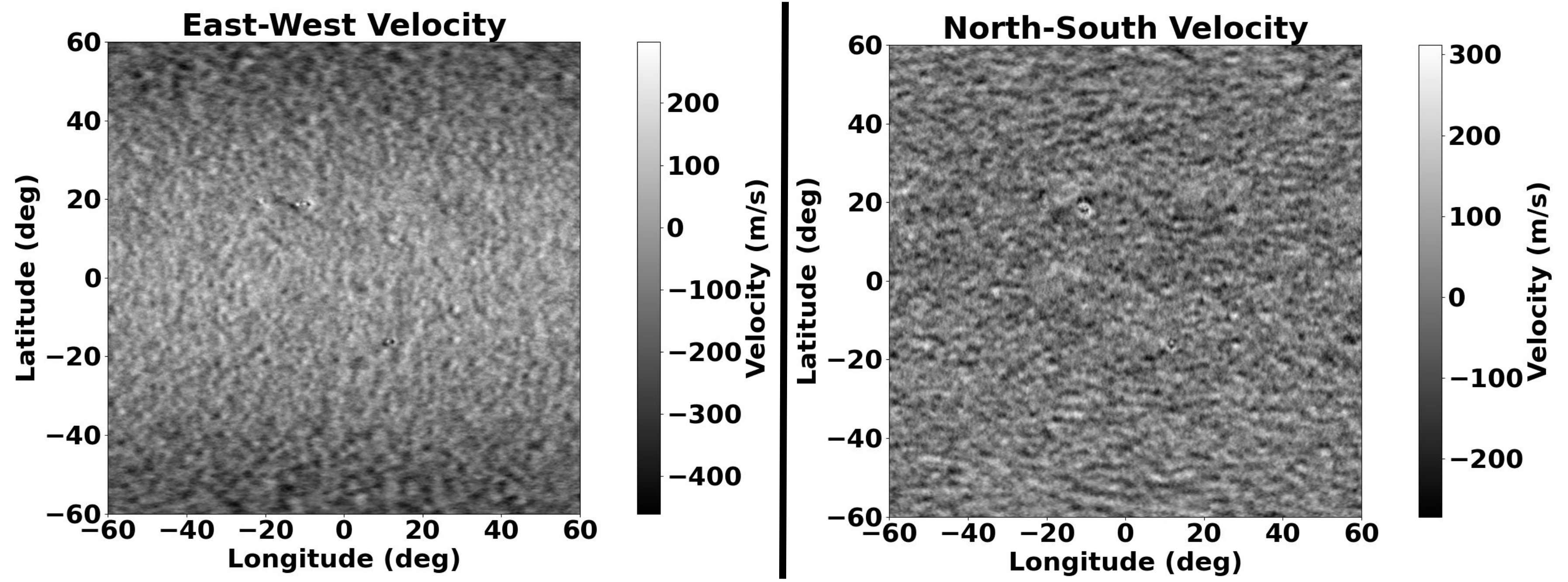
During my 10 weeks at NAS, I have studied more about solar dynamics and experimented with various computer vision methods on this newly available data to develop a digital image processing pipeline that effectively segments supergranules, which are represented as positive divergence values on horizontal flow divergence maps.

Data Used

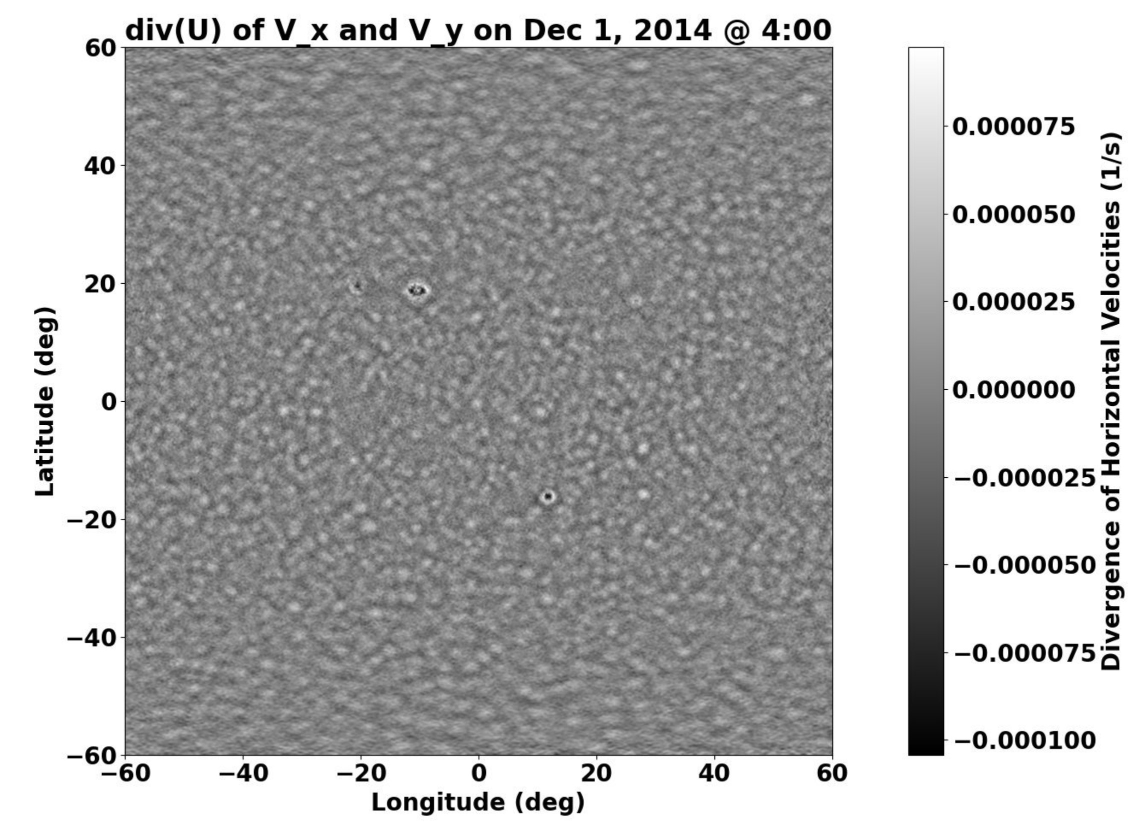
- **Horizontal Velocity and Preprocessed Horizontal Flow Divergence Maps at 1-3 Mm, and Magnetograms from Dr. Junwei Zhao at Stanford University**
    - Zhao's primary inputs were dopplergrams with a selected area that is tracked to remove solar rotation and remapped into heliographic coordinates using Postel's projection relative to the given area's center.
    - 25 regions are selected, for which the East-West velocity (v_x) and North-South velocity (v_y) in each depth layer is derived with a horizontal spatial sampling of 0.12 deg/pixel. Each region's inversion results are obtained in Postel-projection coordinates and then converted into longitude-latitude coordinates and applied with Cubic spline interpolation. High-latitude regions are oversampled. The final result gives us a 1026×1026 image covering 120 degrees in both longitude and latitude.

Horizontal Velocity Maps

(Depth of 1-3 Mm)



Computed div(U) of Horizontal Velocities



Methodology

Image Processing Pipeline

1. East-West and North-South Horizontal Velocity Maps
2. Compute div(U) of Horizontal Velocities
3. Crop and Grayscale
4. Normalization

1. Gaussian Denoising
2. Contrast-Limited Adaptive Histogram Equalization
3. Manual Thresholding
4. Morphological Opening

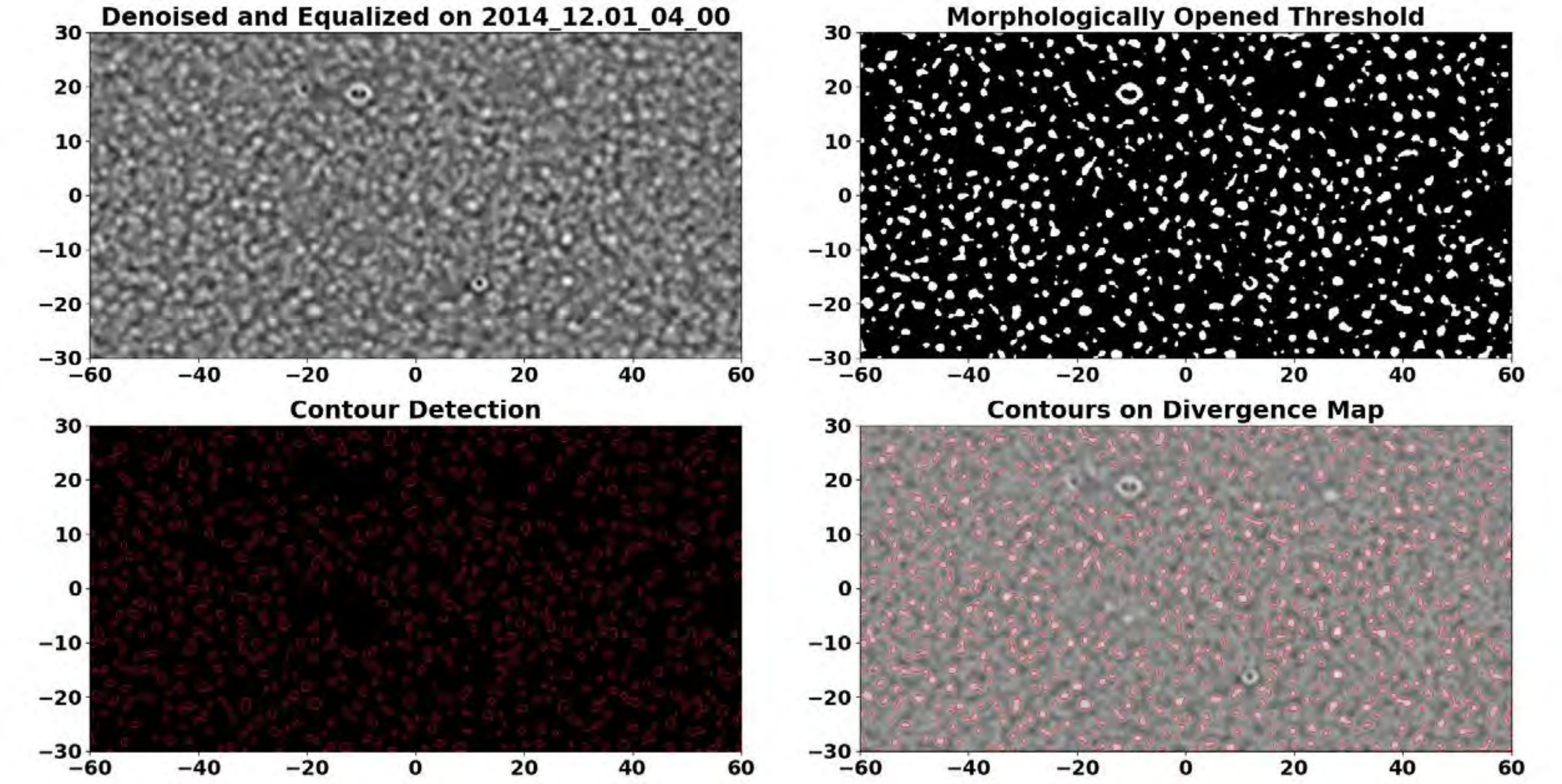
1. Laplacian of Gaussian Blob Detection
2. Masking Supergranules in Active Regions
3. Extract and Plot Contour Properties Over Time

How Can We Detect Supergranules?

Method 1: Contour Detection

- Completely dependent on thresholding.
- Adaptive thresholding algorithms like Otsu did not work very well.
- Could not find an effective normalization to allow all images to apply a common manual threshold.
- Generally, pure thresholding with contour detection either yielded many under-segmented supergranule clusters, or if we tried to increase the threshold to account for under-segmentation, we would lose smaller supergranules and produce too small supergranules compared to the actual area shown on the divergence map.

Method 1: Contour Detection



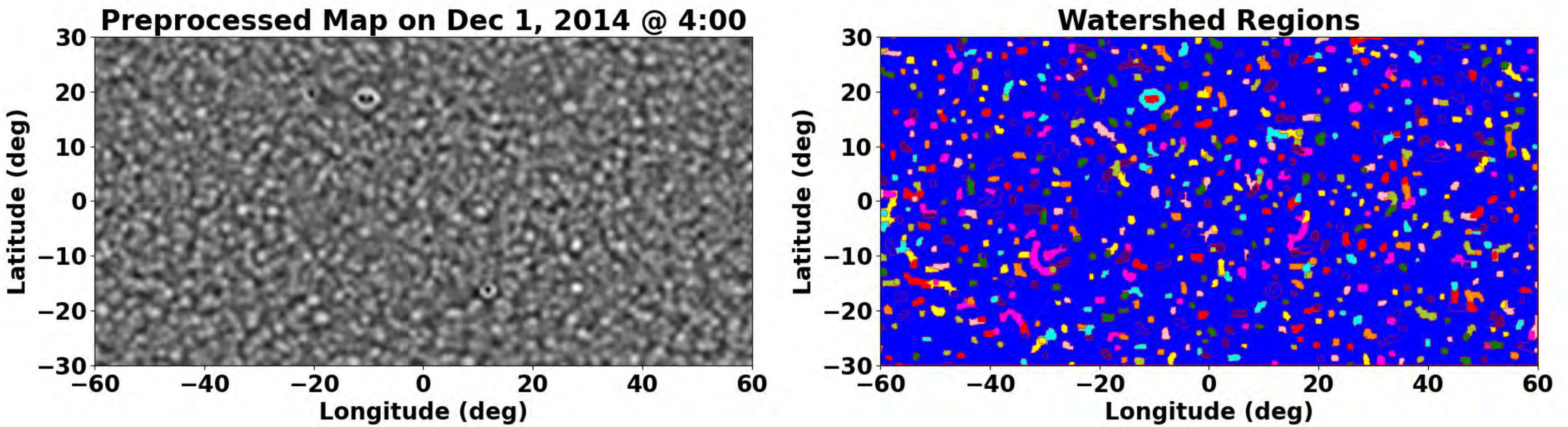
Method 2: Watershed Segmentation

1. Threshold the image to be binary.
2. Dilate the image to create the "sure background" mask.
3. Calculate the distance transform of the image, which provides local minima of where the supergranules are by assigning values to pixels based on their distance from the nearest "1" value in the binarized image. For example, 1's would be assigned as 0.
4. Manually threshold the distance transform to create the "sure foreground" mask, which will be labeled for every individual region and become your marker template.
5. Calculate the Unknown Regions by (sure background - sure foreground).
6. To create markers for the Watershed Algorithm to detect, add 20 to all values and assign a value of 0 wherever there is an unknown region. This allows us to see the boundaries of the sure foreground, sure background, and unknown regions on one image.
7. Use these markers to apply Watershed Algorithm to compute Watershed Regions.

Method 2: Watershed Segmentation

- Known to be effective at separating cell clusters in biomedical imaging. Could it be useful for us?
- Could not fine-tune the distance transform threshold to generalize to all images.
- Some images produced better Watershed regions than others. Overall, I didn’t have success with this. Although, other research papers did using a different implementation.

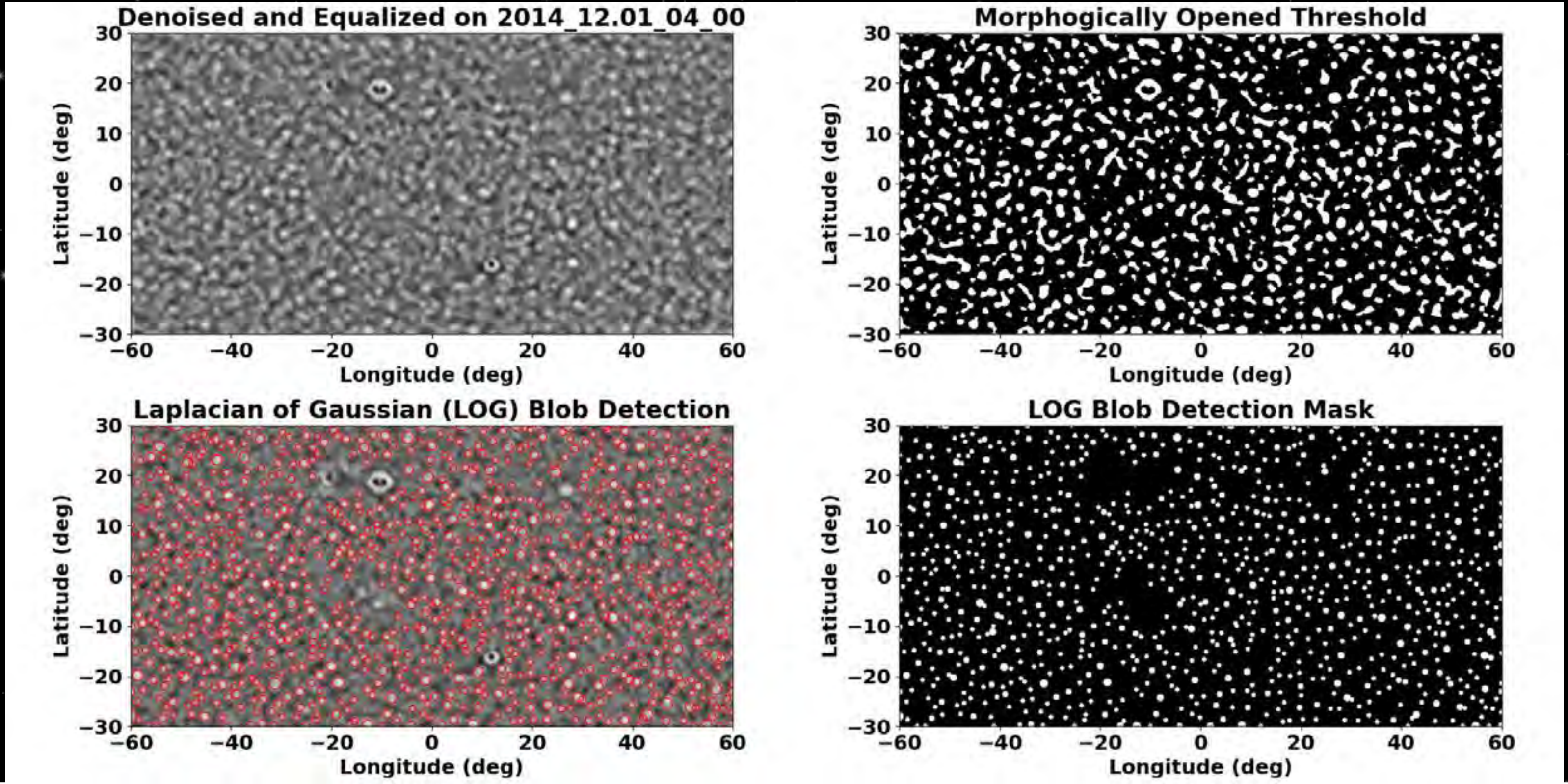
Method 2: Watershed Segmentation



Method 3: Laplacian of Gaussian Blob Detection

- Tried SciKit LoG Blob Detection algorithm since I could not find an effective normalization for all images to apply a common threshold for contour detection. I suspected Blob Detection can be a workaround.
- Advantages:
  - Found that it can detect small variations in thresholded supergranule clusters and segment them fairly well. Also, it can segment very blurry supergranules if I use a threshold of 155-255 pixel intensity on the preprocessed divergence maps.
- Disadvantages:
  - Only outputs blob center (x, y) and radius of blob. Assumes all supergranules to be circular.

Laplacian of Gaussian Blob Detection

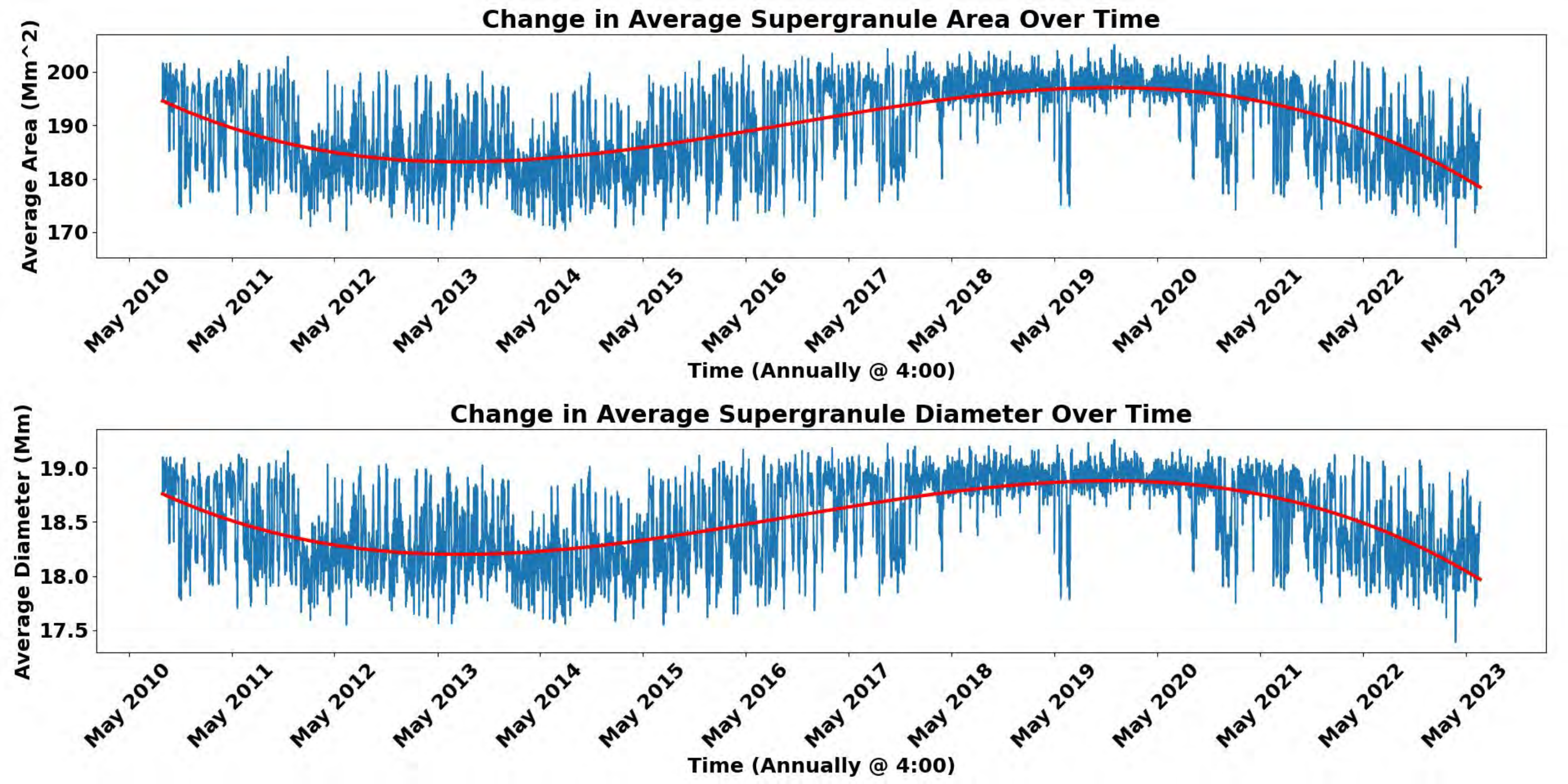


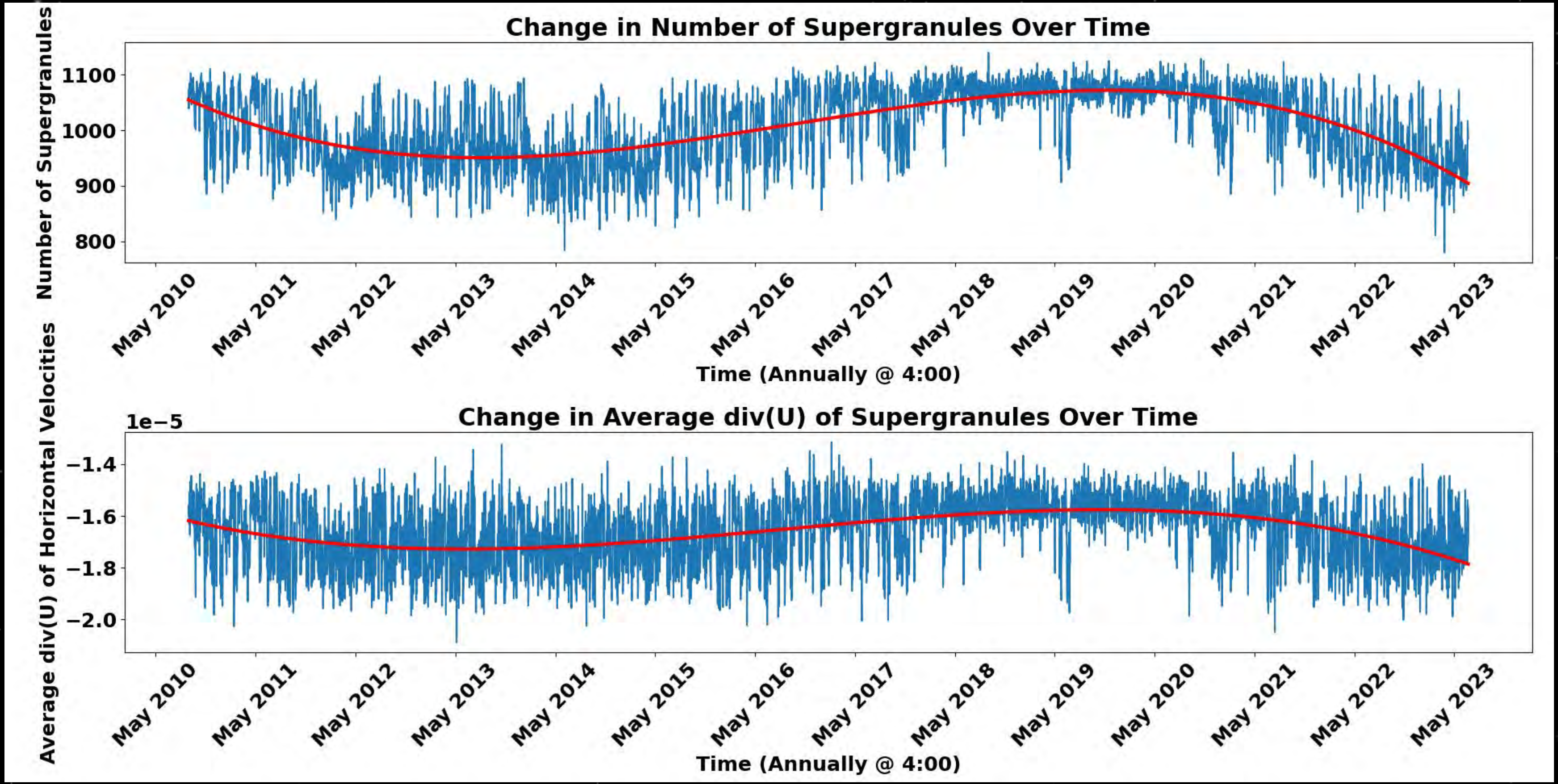
Masking Segmentations in Active Regions

- Active regions often exhibit high divergence values which can be mistaken as supergranules.
- To remove misidentified supergranules in active regions, I identified all magnetogram coordinates with an absolute value of 1,000 Gauss and over.
- I specified a radius around each coordinate, essentially a bounding circle.
- If a segmented supergranule is within this region, then we remove it.

Results

Using LoG Blob Detection

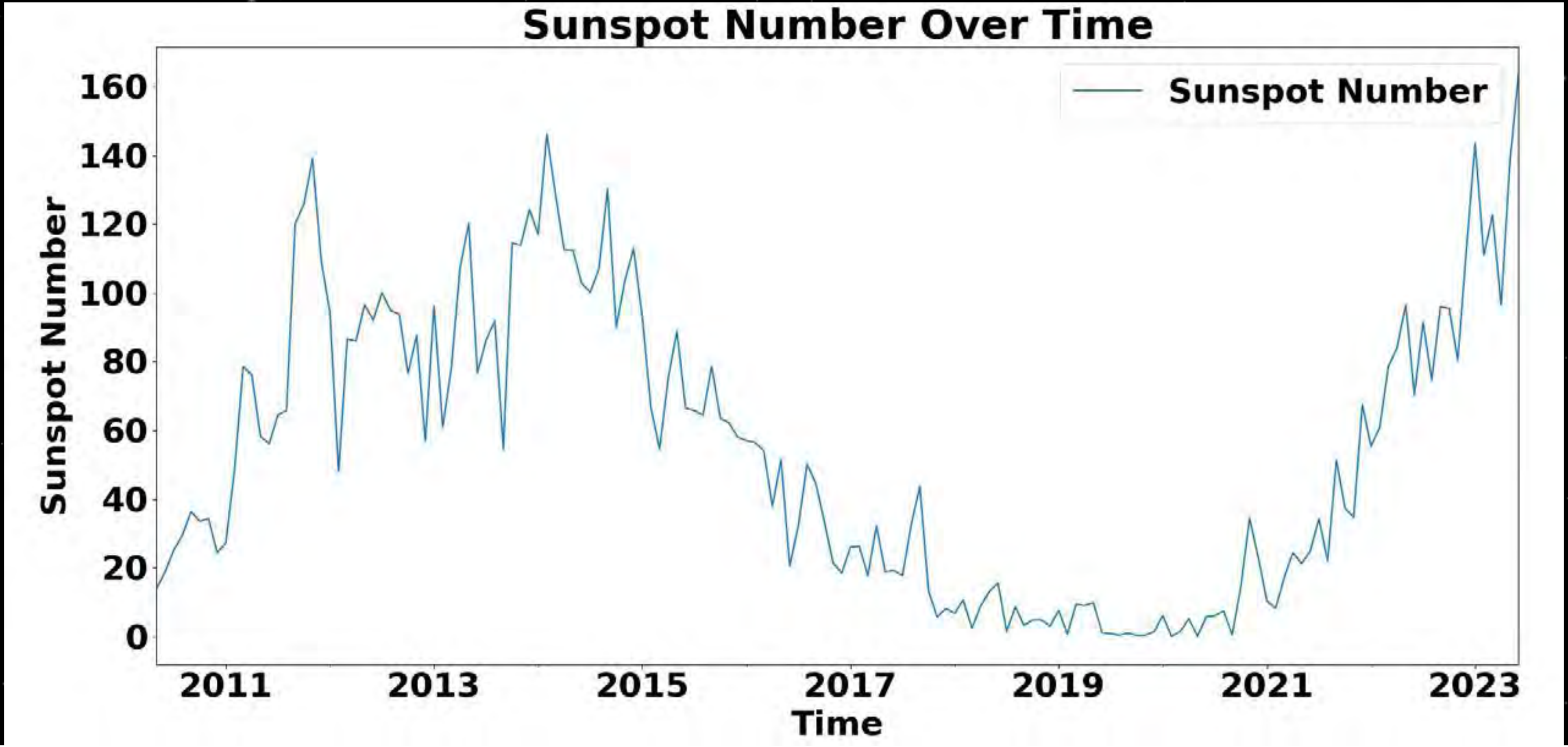

Change in Average Supergranule Area Over Time
Average Area (Mm^2)
200
190
180
170
May 2010
May 2011
May 2012
May 2013
May 2014
May 2015
May 2016
May 2017
May 2018
May 2019
May 2020
May 2021
May 2022
May 2023
Time (Annually @ 4:00)
Change in Average Supergranule Diameter Over Time
Average Diameter (Mm)
19.0
18.5
18.0
17.5
May 2010
May 2011
May 2012
May 2013
May 2014
May 2015
May 2016
May 2017
May 2018
May 2019
May 2020
May 2021
May 2022
May 2023
Time (Annually @ 4:00)



Change in Number of Supergranules Over Time
Number of Supergranules
1100
1000
900
800
May 2010
May 2011
May 2012
May 2013
May 2014
May 2015
May 2016
May 2017
May 2018
May 2019
May 2020
May 2021
May 2022
May 2023
Time (Annually @ 4:00)
Change in Average div(U) of Supergranules Over Time
Average div(U) of Horizontal Velocities
1e−5
−1.4
−1.6
−1.8
−2.0
May 2010
May 2011
May 2012
May 2013
May 2014
May 2015
May 2016
May 2017
May 2018
May 2019
May 2020
May 2021
May 2022
May 2023
Time (Annually @ 4:00)

Sunspot Number Over Time

(Data source: https://www.sidc.be/SILSO/datafiles)



Analysis

- Firstly, it's important to note that the LoG Blob Detection method produced diameters in the ranges of 17.5 - 19.0 Mm, which is different than the typically researched 20 - 30 Mm diameter.
- Secondly, both Contour Detection and Blob Detection methods produce similar curves.
  - Supergranule number, area, and diameter tends to increase as sunspot number decreases and vice versa.
- This aligns with the following research paper's results by N. Meunier, T. Roudier, and M. Rieutord: https://www.aanda.org/articles/aa/pdf/2008/36/aa8835-07.pdf
  - They also found that supergranule area tends to decrease as sunspot number increases and when active regions display higher activity.

Obstacles and Lessons Learned

- Reproducing divergence maps with correct divergence values of the horizontal velocities.
- At first, I considered training a convolutional neural network model.
    - Spent first couple weeks partitioning image with padding to maintain resolution, initial threshold labeling, and manually fixing these labels.
    - However, I realized image processing alone may suffice and would be simpler, cheaper, and more computationally efficient.

Obstacles and Lessons Learned

- How to segment supergranule clusters? → Tried Watershed Segmentation and Blob Detection. Blob Detection seems to work well.
- Removing active regions misidentified as supergranules → Use magnetograms to mask out regions with absolute values of 1,000 Gauss and above.
- Learned about horizontal velocity fields, supergranulation, and active regions. Also learned more about computer vision methods and SciKit and OpenCV libraries.
- I improved my approach towards starting projects, experimenting, and explaining my work to a team.

Conclusion

- My pipeline runs quite slowly (4-5 seconds for each horizontal divergence map). If it continues to be used, we can possibly optimize it with vectorization and parallelization in the future.
- If needed, we can explore more deep learning techniques in the future, such as semantic segmentation for this problem.
- My team was so diverse and inspiring, and this internship was an outstanding learning experience. I explored so much about astronomy and computer vision and plan on using what I learned for future ML projects!

Acknowledgements

Special thank you to...

Dr. Junwei Zhao, Stanford University, for providing his horizontal velocity and divergence maps, magnetograms, and research paper with information about his computed divergence map to guide us in computing our own and helping me learn about solar properties and supergranules.

Dr. Irina Kitiashvili for coordinating meetings, providing guidance about supergranules, active regions, computing the divergence map, and organizing files on Pleiades.

Dr. Spyros Kasapis and his colleagues for providing guidance on machine learning and computer vision methods.

Emma Opper, our previous intern who worked on this project for her initial segmentation algorithm and documentation on retrieving data from the FITS files.