Decoding the Sun

Author's Name: Jiang Wu

Coach Name: Debra Dimas

Host Organization: Stanford

ETP Type: Classroom

Subject/Grade: Computer Science / Grade 11

Cost: Low

Length of Lesson: Multi Day

Abstract (~150 words)

This project introduces high school students to programming concepts through writing a Python script that finds and counts sunspots in real images of the Sun. Across five class periods, students work in pairs and learn about how images are processed by a computer, how to create and traverse a 2D array, and write a complete sunspot detection program. This requires setting brightness cutoffs to separate the dark spots from the bright surface, tagging the pixels that qualify, grouping the touching pixels into individual spots, and filtering out specks too small to count.

In addition to programming concepts, students will also learn to justify their choices. Because deciding how dark or big a sunspot must be and where its edges lie has no objectively correct answer, each student's decisions when writing their program become arguments they must explain. Students then validate their automated counts against a human count by their peers, and analyze why the two differ.

Focal Content & Supporting Practices

All three standards this project focuses on are taken from the California Computer Science State Standards:

- 9-12.DA.8: Translate between different representations of data abstractions of real-world phenomena, such as characters, numbers, and images.
- 9-12.DA.10: Create data visualizations to help others better understand real-world data.
- 9-12.DA.11: Refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.

21st Century/Durable Skills and Applications (1 - 2 bullets)

- Data literacy: Students learn how to read and work with data
- Technical problem solving: Students break a goal down into a sequence of solvable steps

Measurable Objective(s)

- Students will convert an intensitygram of the Sun into a numeric 2D array based on the brightness of each pixel.
- Students will build a program that calculates the number of sunspots in an intensitygram of the Sun.
  - Note from mentors: Advanced students can challenge themselves using daily magnetograms (https://jsoc1.stanford.edu/hmi_latest) which provides more information than the intensitygram
- Students will justify their intensity threshold and minimum sunspot size choices. The intensity threshold is the minimum absolute value difference between a pixel and its neighbor to count as being part of a sunspot.

Formative Assessment(s)

- Daily exit tickets:
    - How is this image like a spreadsheet?
    - What does a pixel value close to 0 mean? What about close to 255?
    - What happens when you make the threshold stricter? What about more relaxed?
    - What happens to your sunspot count as you increase the minimum size?
- General questions I'll ask students as they're pair programming:
    - What is the goal of the code your partner is typing?
    - What edge cases does your program check for?
    - If I notice students debugging: What do you think caused the error?
    - What do you expect this line to do?

Summative Assessment(s)

- Each student submits their commented code, with justifications of their intensity threshold and minimum size choices. Students will be provided a rubric to evaluate their code.

Fellowship Description (300-500 words)

My fellowship is hosted by the Kavli Institute for Particle Astrophysics and Cosmology (KIPAC), a laboratory run jointly by Stanford university and the SLAC National Accelerator Laboratory. Within KIPAC, I work with the Solar Physics Group, which studies our Sun. One of the group's central goals is to understand the Sun's magnetic fields and flows, and this is where my project comes in.

The Sun's magnetism rises and falls on a roughly 11-year cycle, and sunspots are the places where especially strong magnetism appears. One way to study how this magnetism behaves is to use a computer program known as a surface flux transport that simulates the magnetic field across the Sun's surface. The program starts with a map of the Sun's surface, then steps forward in time, calculating at each step how the magnetism moves and spreads. It accounts for how the sun spins unevenly, how gas drifts from the equator toward the poles, and how magnetism naturally spreads out.

The solar flux transport model given to me was written in an older programming language called Fortran. My primary task is to translate it into Python. The reason for this is that relatively few people learn Fortran today, and in contrast Python is one of the most popular programming languages. It is also easier to read, comes with powerful libraries for data analysis and visualization, and is far more accessible to researchers and students. A key challenge of this project is ensuring the Python version produces exactly the same results as the original Fortran version.

Once the simulation runs in Python, I'll use its output to create graphs that reveal patterns in the data, like the butterfly diagram. This plots the sine of the latitude at which the sunspots appear against time. I'll then need to produce an animation that projects the actual movement of the sunspots onto an image of the Sun's surface.

Fellowship Connection to School/Classroom (300-500 words)

The school I work at is part of the New Tech Network, which has project-based learning as one of its pillars. One of the most valuable aspects of this fellowship is that it connects computer science to a real-world scientific problem. The project investigates how sunspots emerging on the solar surface affect the Sun's large-scale magnetic field. This is an ideal classroom connection because sunspots are visible, concrete phenomena and are an ideal candidate for data analysis lessons. Students often ask "When are we going to use this in real life", and having worked on this project, I can show students that coding is not only used to build apps or games, but also as a tool for asking scientific questions, testing ideas, and making sense of complex systems.

The fellowship also helps me create stronger project-based learning experiences. During the research project, I change simulation inputs, run many trials with sunspots in different locations, and analyze how the results change. This process can be translated directly into the classroom. Students could be asked to change variables, run simulations, collect results, graph data, and explain the patterns they observe. This would help students understand important computational thinking concepts such as variables, parameters, iteration, abstraction, and causal relationships.

The simulation component is especially powerful because it mirrors the way professional scientists and researchers work. In school, students are often given problems with known answers. In research, the answer is not known in advance. This fellowship gives me firsthand experience with that, which helps me create a classroom culture where students are comfortable with revising their work, comparing methods, and explaining why their results may differ from someone else's. For example, students analyzing solar images might choose different thresholds for what counts as a sunspot.

Participating in this fellowship has already improved my skills with data visualization. I was taught how to use Matplotlib, a Python tool for making graphs. This is great because graphs make patterns easier to notice. I believe learning Matplotlib will make computer science feel more real for students because they can see the results of their code. This will also aid students when presenting their work to others.

Instructional Plan (This is the bulk of your ETP and may take several pages.)

Decoding The Sun Lesson Plan

Day	**Focus**	**Code students write (from the starter file)**
1	What is a sunspot? + teacher demo + meet the code	Nothing yet
2	An image is a grid of numbers	`brightness_grid`, `find_brightest_pixel`, `median`
3	Finding the Sun and its "normal" brightness	`count_sunspots` Steps 1–3
4	Marking spots, painting them pink. The first full run	`count_sunspots` Step 4, `save_highlighted_image`, the info loop in `__main__`
5	Run on the live Sun, fine tune, compare to real data	Complete anything that hasn't been done yet

Additional Supports

Tools to meet the needs of all learners (UDL, SEL, ELL, SPED)

Key Vocabulary Scaffold (primarily for EL students to translate key vocabulary words)
Fill In The Blanks Scaffold (scaffold where the big ideas are presented in fill in the blank form with an answer bank)
Parsons Problems Scaffold (high level of support where the correct code is already provided, but in a scrambled order. Students need to rearrange them into the correct order)

Materials

Include links to all files within this ETP

Link to ETP Folder
Link to lesson plan
Slideshow that accompanies this lesson
Starter Code
Decoding the Sun Rubric
Scaffold Files:

- Key Vocabulary Scaffold
- Fill In The Blanks Scaffold
- Parsons Problems Scaffold

For teacher reference: sunspots.py (for teacher use, this is to demo the finished project so students know what they're working towards)
Daily intensitygram of the Sun (for students)
Other visualizations of the Sun for advanced students
Other visualizations of the Sun for advanced students (part 2)

- Teacher station with projector or smartboard
- Student computers, one per student or pair
- Reliable internet access
- Python 3
- VS Code (or similar IDE)

References

Bridgman, T. (2012, November 20). *The active Sun from SDO: HMI magnetogram* [Video]. NASA Scientific Visualization Studio. https://svs.gsfc.nasa.gov/3989

National Aeronautics and Space Administration. (n.d.). *Latest HMI intensitygram (1024 × 1024)* [Image]. Solar Dynamics Observatory. https://sdo.gsfc.nasa.gov/assets/img/latest/latest_1024_HMII.jpg

Rempel, M., & Schlichenmaier, R. (2011). Sunspot modeling: From simplified models to radiative MHD simulations. Living Reviews in Solar Physics, 8, 3. https://doi.org/10.12942/lrsp-2011-3

Solar Monitor. (2026). *Solar activity for 9 May 2026* [Data set]. Dublin Institute for Advanced Studies, Solar Physics Group. https://solarmonitor.org/

Keywords (2-4)

Sun Computer Science Data