DEEP LEARNING SCHOOL

Solar Orbiter Images Challenge

Group Name: HelioMind

Group Members

1. Anastasiia Omeliukh
2. Hitesh Abichandani
3. Prateek
4. David Okpaga
5. Eugenio Vitale

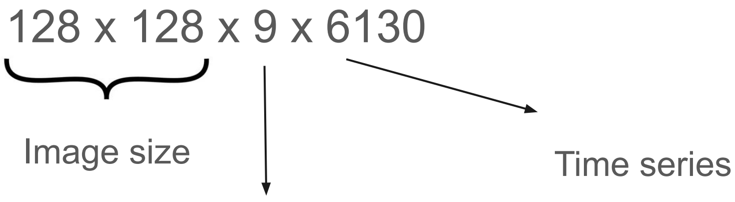
ERUM DATA HUB

May 9, 2025

Organization Among the Team

(self-organized) **chaos**

Dataset

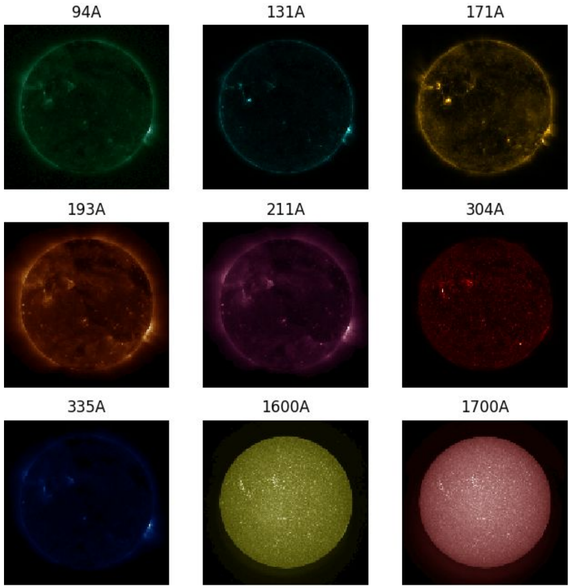


128 x 128 x 9 x 6130

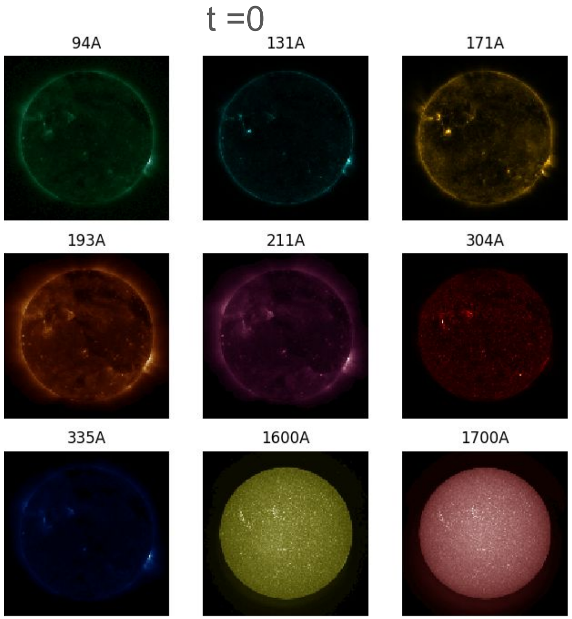
Image size

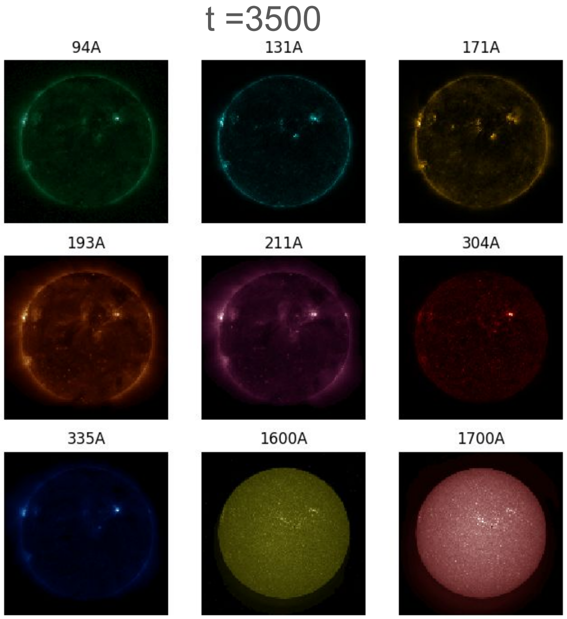
Time series

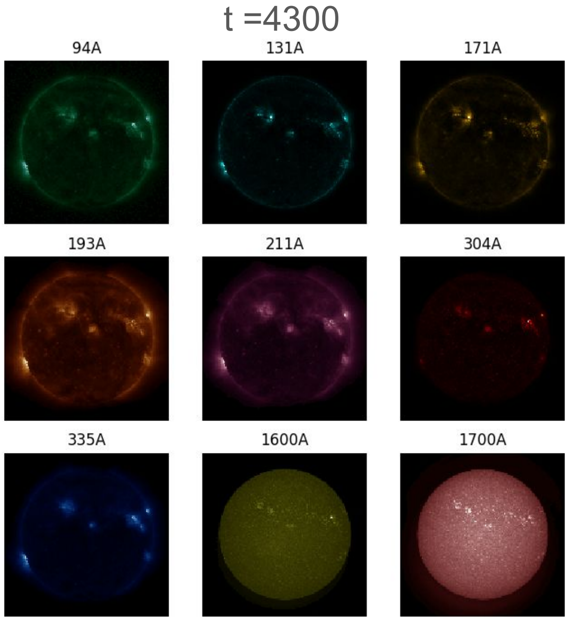
Channels corresponding to different wavelegths



Note on correlations between channels

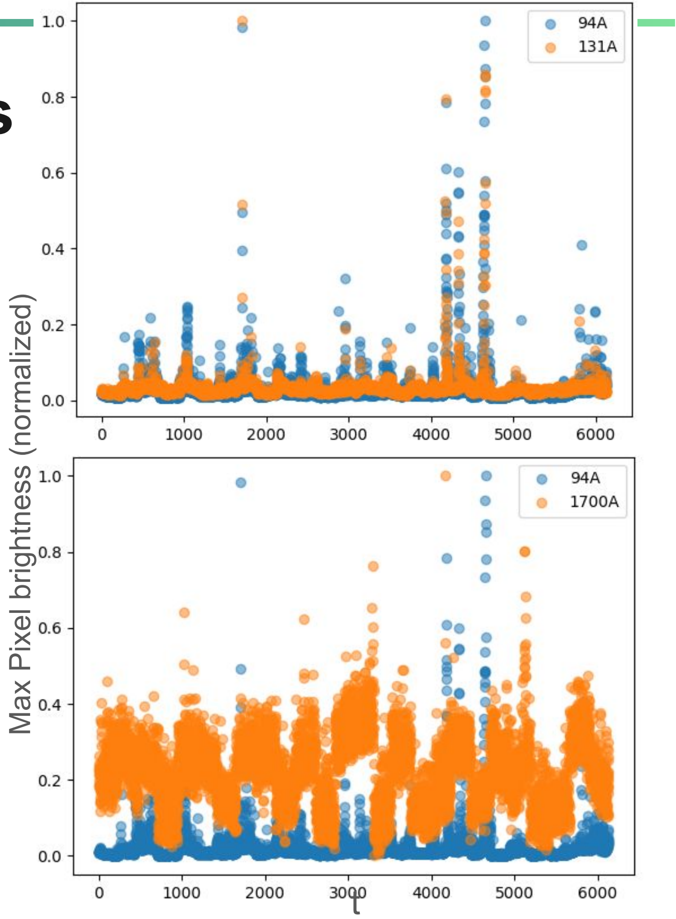






Note on correlations between channels

- not all channels are correlated
    - Likely indicates different origin of radiation



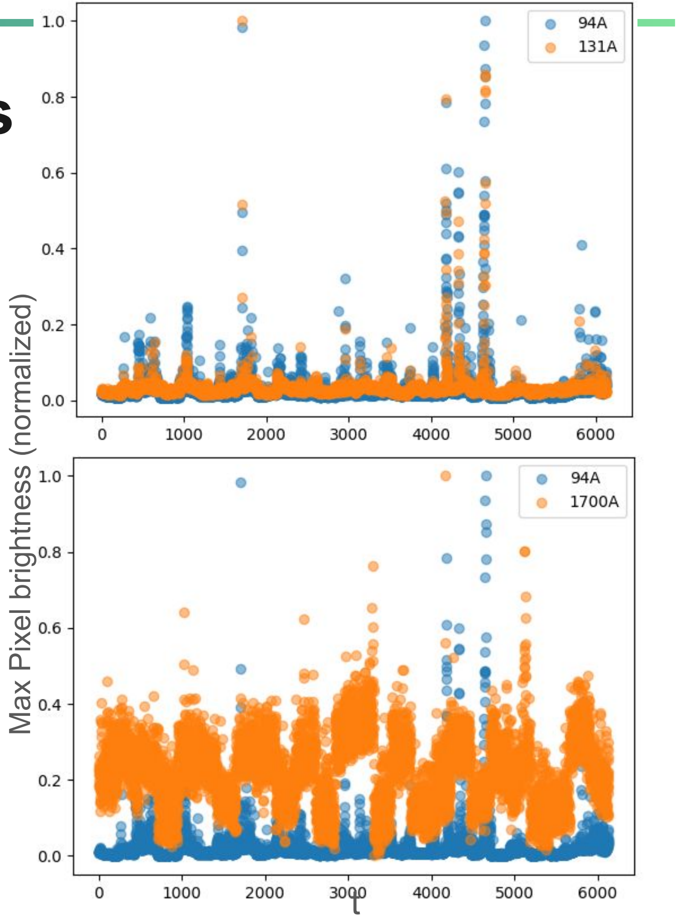
Note on correlations between channels

- not all channels are correlated
  - Likely indicates different origin of radiation

AIA (Atmospheric Imaging Assembly)

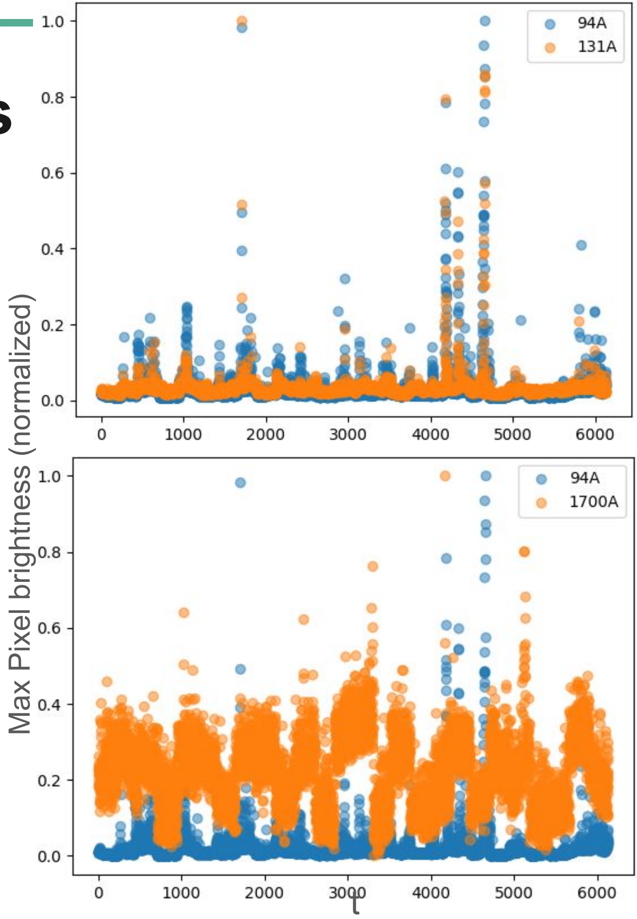
(from the challenge notebook)

- Observes Chromosphere, Transition Region, and Corona
- 10 different wavelenghts (in this challenge only 9)
- Investigates mechanisms that heat the corona



Note on correlations between channels

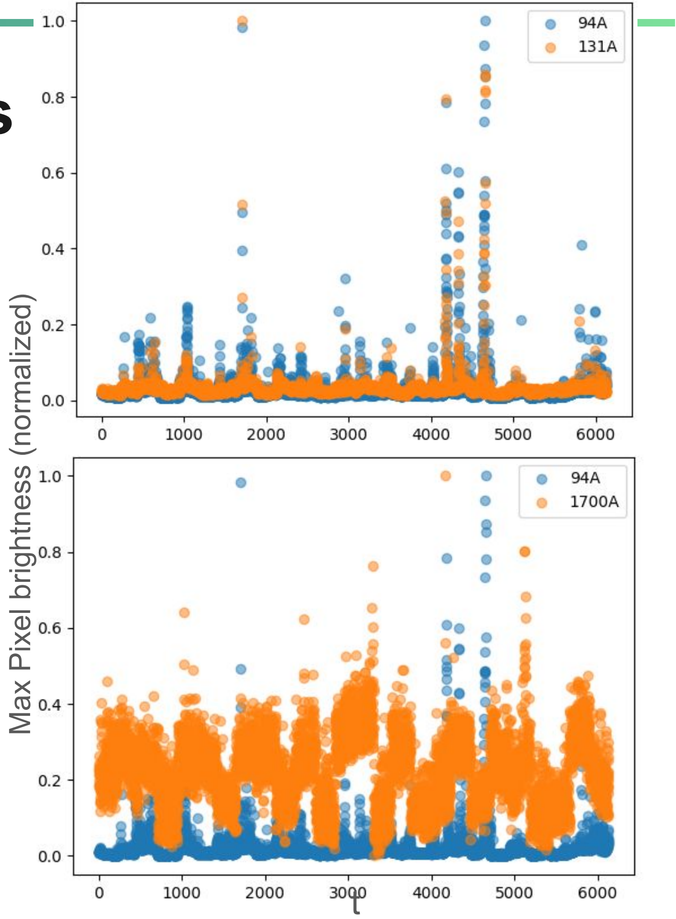
- not all channels are correlated
  - Likely indicates different origin of radiation
  - a-priori cannot yield an accurate prediction



Note on correlations between channels

- not all channels are correlated
  - Likely indicates different origin of radiation
  - a-priori cannot yield an accurate prediction

- idea: study time-correlation between the channels and select only data periods with stronger correlation
  - cannot predict flares
  - allows us to predict “quiet” state with higher accuracy



Input data and target

Shape of input data: (6130, 128, 128)
Shape of target: (8, 6130, 128, 128)

batch_size = 64

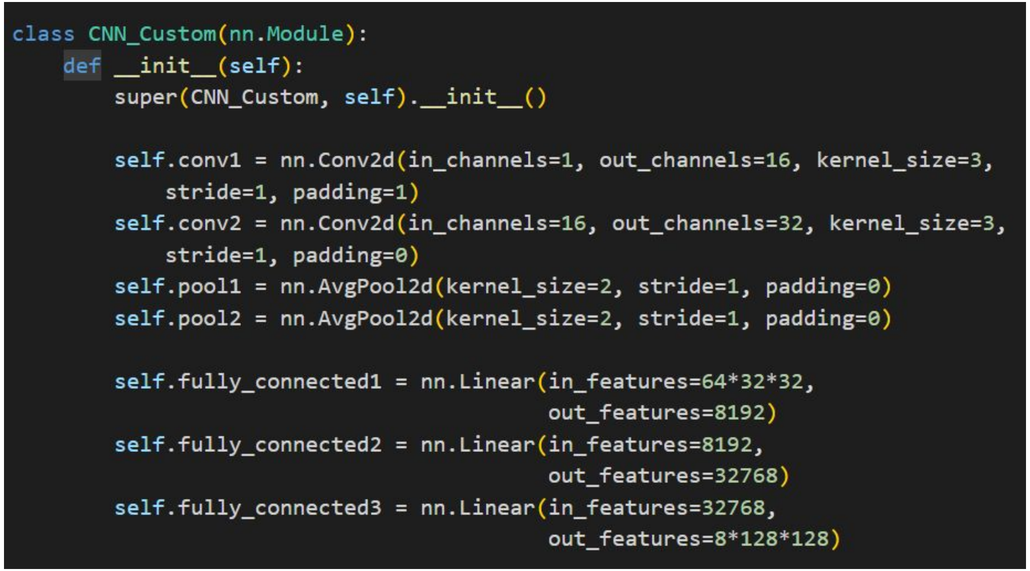
Dataset = CustomDataset(input_data, target)

Random Split (80%, 10%, 10%)

DataLoader for mini-batching for Efficiency

- Input Batch: torch.Size([batch_size, 128, 128])
- Target Batch: torch.Size([batch_size, 8, 128, 128])

Model Idea

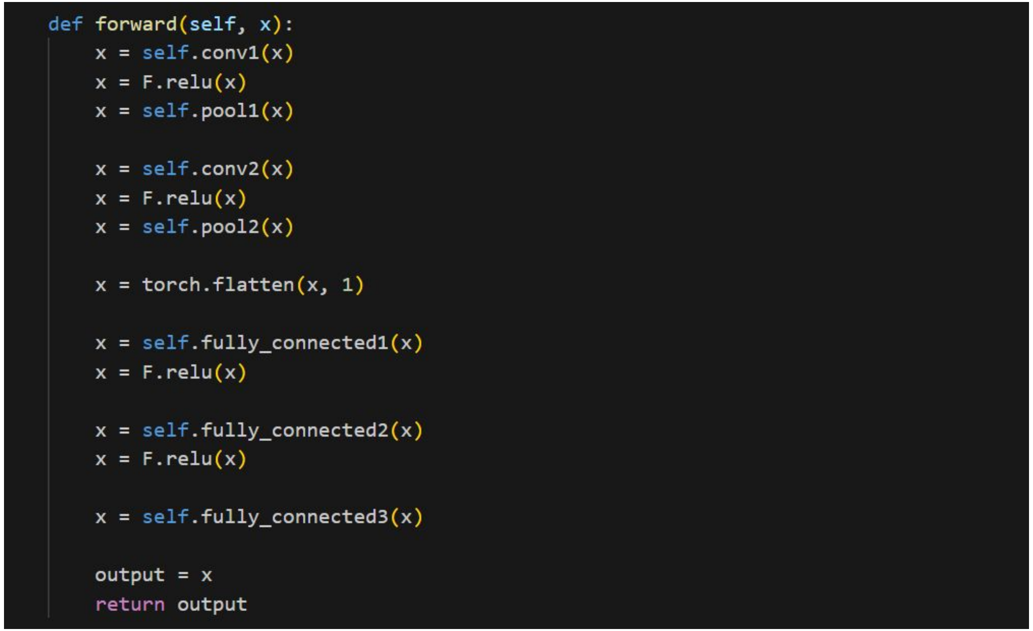


```
class CNN_Custom(nn.Module):
    def __init__(self):
        super(CNN_Custom, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=1, out_channels=16, kernel_size=3,
            stride=1, padding=1)
        self.conv2 = nn.Conv2d(in_channels=16, out_channels=32, kernel_size=3,
            stride=1, padding=0)
        self.pool1 = nn.AvgPool2d(kernel_size=2, stride=1, padding=0)
        self.pool2 = nn.AvgPool2d(kernel_size=2, stride=1, padding=0)

        self.fully_connected1 = nn.Linear(in_features=64*32*32,
                                          out_features=8192)
        self.fully_connected2 = nn.Linear(in_features=8192,
                                          out_features=32768)
        self.fully_connected3 = nn.Linear(in_features=32768,
                                          out_features=8*128*128)
```

Model Idea



```python
def forward(self, x):
    x = self.conv1(x)
    x = F.relu(x)
    x = self.pool1(x)

    x = self.conv2(x)
    x = F.relu(x)
    x = self.pool2(x)

    x = torch.flatten(x, 1)

    x = self.fully_connected1(x)
    x = F.relu(x)

    x = self.fully_connected2(x)
    x = F.relu(x)

    x = self.fully_connected3(x)

    output = x
    return output
```

Results

Environmental Impact

- Training did not converge successfully despite multiple architect attempts - Deconvolution, LNET
- We trained 350 Grams of CO2 emission tracking

Outlook

- We analyzed the data, tried to build a model, but didn’t succeed
- We look forward to getting more ideas on how to improve our approach!