Image Processing Methods for Coronal Hole Segmentation, Matching, and Map Classification

V. Jatla, *Member, IEEE,* M.S. Pattichis, *Senior Member, IEEE,* and C.N. Arge

***Abstract*—The paper presents the results from a multi-year effort to develop and validate image processing methods for selecting the best physical models based on solar image observations. The basic approach consists of selecting the physical models based on their agreement with coronal holes extracted from the images. Here, we note that coronal holes are associated with open magnetic fields regions that produce much of the solar wind. High speed wind emerging from deep within the corona can disrupt the Earth's magnetic field generating geomagnetic storms.**

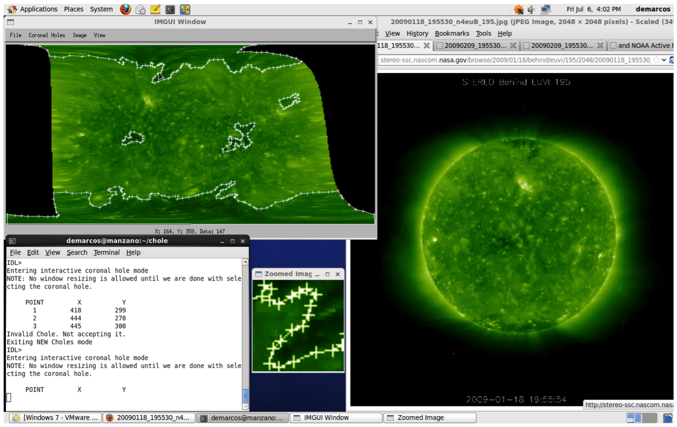
We decompose the problem into three subproblems: (i) coronal hole segmentation based on physical constraints, (ii) matching clusters of coronal holes between different maps, and (iii) physical map classification. For segmenting coronal holes, we develop a multi-modal method that uses segmentation maps from three different methods to initialize a level-set method that evolves the initial coronal hole segmentation to the magnetic boundary that marks the transition from one polarity to another. Then, we introduce a new method based on Linear Programming for matching clusters of coronal holes. The final matching is then performed using Random Forests.

The methods were carefully validated using consensus maps derived from multiple readers, manual clustering, manual map classification, and method validation for 50 maps. The proposed multi-modal segmentation method significantly outperformed SegNet, U-net, Henney-Harvey, and FCN. Overall, the method gave a 95.5% classification accuracy.

***Index Terms*—Solar images, Consensus maps, Segmentation, Matching, Classification, Random Forests.**

I. Introduction

Intense solar activity can cause severe disruptions to the Earth's magnetic field. Typically, open magnetic field lines on the sun form the origin of solar wind that reaches the earth. Forecasting requires accurate physical modeling of the sun's magnetic field and accurate tracking of open magnetic field lines. Solar regions characterized by open magnetic lines are associated with coronal holes. Thus, in summary, to support forecasting, our goal is to develop solar image analysis methods that can automatically segment coronal holes in solar observations, match them to coronal holes generated by candidate physical models, and then select the best candidate models for forecasting.


Applications Places System
Fri Jul 6, 4:02 PM demarcos
IMGUI Window
File Coronal Holes Image View
X: 164, Y: 359, Data: 147
20090118_195530_n4euB_195.jpg (JPEG Image, 2048 × 2048 pixels) - Scaled (34
View History Bookmarks Tools Help
stereo-ssc.nascom.nasa.gov/browse/2009/01/18/behind/euvi/195/2048/20090118_195530
STEREO Behind EUVI 195
2009-01-18 19:55:54
http://stereo-ssc.nascom.nasa
Zoomed Imag
[Windows 7 - VMware ...
20090118_195530_n4...
demarcos@manzano:...
IMGUI Window
Zoomed Image

demarcos@manzano:~/chole

File Edit View Search Terminal Help

```
IDL>
Entering interactive coronal hole mode
NOTE: No window resizing is allowed until we are done with sele
cting the coronal hole.

     POINT         X           Y
       1          418         299
       2          444         270
       3          445         300
Invalid Chole. Not accepting it.
Exiting NEW Choles mode
IDL>
Entering interactive coronal hole mode
NOTE: No window resizing is allowed until we are done with sele
cting the coronal hole.

     POINT         X           Y
```

Fig. 1: The coronal hole segmentation problem. The figure illustrates the manual segmentation process where the coronal holes appear as dar regions in the Extreme ultraviolet (EUV) images. Black regions represent solar regions for which we do not have any observations.

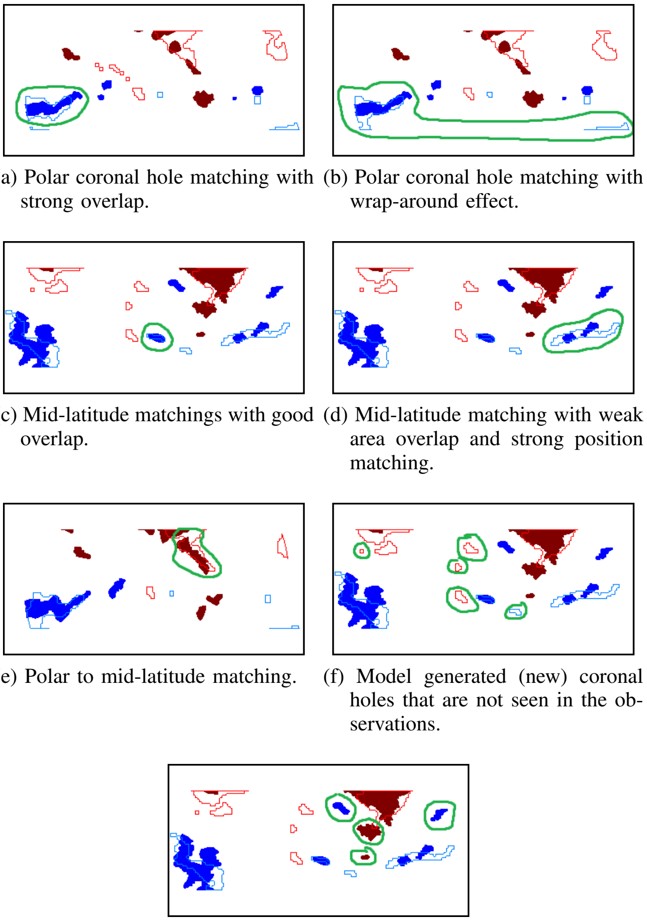
We present an example of the coronal hole segmentation problem in Fig. 1. Coronal holes are distinguished as regions of the same magnetic polarity, lower electron density, and lower temperatures [1]. In the extreme ultraviolet (EUV) region, coronal holes appear darker than their surrounding regions (see Fig. 1).

Physical models of the corona and solar wind include Potential Field Source Surface based models, such as the Wang-Sheeley-Arge (WSA) model [2] and the magnetohydrodynamics (MHD) models such as CORHEL [3]. Physical models use solar image observations of the photospheric magnetic field as their input to model the solar wind propagating throughout the heliosphere. Unfortunately, our physical models are incomplete. Certain parameters are missing. Examples of missing parameters include (a) the source surface height in the Potential Field Source Surface (PFSS) model (see references in [2]) of the solar atmosphere (corona), and (b) the amount of heating in more advanced Magnetohydrodynamics (MHD) coronal models. In what follows, we assume the use of the WSA model based on PFSS. In WSA, the amount of open flux and hence the coronal hole area provided by the physical model depends on the source surface radius used (i.e., it is a free parameter in the model beyond which all magnetic field lines flow out into space forming the solar wind). More advanced MHD models estimate the amount of heating that will in-turn affect the amount of open flux and hence the coronal hole area [4]–[6].

To assess the performance of a map produced by a model, there is a need to establish ground truth from solar observations. Unfortunately, the standard practice based on segmenta-

V. Jatla and M.S. Patthics are with Department of Electrical and Computer Engineering, The University of New Mexico, Albuquerque, NM 87131. E-mail: (venkatesh369@unm.edu; pattichi@unm.edu)

C. N. Arge is the Chief of the Solar Physics Laboratory in the Heliophysics Science Division at the National Aeronautics and Space Administration's Goddard Space Flight Center of. E-mail: (charles.n.arge@nasa.gov).



(a) Polar coronal hole matching with strong overlap.

(b) Polar coronal hole matching with wrap-around effect.

(c) Mid-latitude matchings with good overlap.

(d) Mid-latitude matching with weak area overlap and strong position matching.

(e) Polar to mid-latitude matching.

(f) Model generated (new) coronal holes that are not seen in the observations.

(g) Example showing several coronal holes missing from the observations.

Fig. 2: Coronal hole cluster matching problem. The thick green lines are used for circling out the different matching scenarios. The consensus maps are represented by solid colored coronal hole maps. Negative/positive-polarity coronal holes are depicted as blue/red. The hollow, light-blue/red curves represent the boundaries of the negative/positive-polarity coronal holes derived from the physical models.

tions by a single human expert can produce significant inter-rater variability. To reduce inter-rate variability, we require that the solar images should first be manually segmented by two independent experts. After manual segmentation, the experts meet and agree upon a consensus map that represents their combined efforts.

To automate the segmentation process, we consider an automatic segmentation approach. We begin with a review of modern segmentation methods and follow-up with a brief summary of our proposed approach. We note that modern segmentation methods have been dominated by neural-network based approaches. We consider recent segmentation methods that are based on convolutional networks for visual image classification. As we shall describe later, we incorporated the best performing methods into our segmentation system to allow for better initialization to our level-set segmentation approach.

In [7], [8], the authors developed fully convolutional networks (FCN) for semantic segmentation. The basic idea is to use transfer learning from popular, pre-trained visual classifiers: AlexNet [9], VGG net [10], and GoogLe Net [11], by adding multi-scale deconvolutional layers to predict ground truth segmentation maps from previously trained convolutional layers. The deconvolutional layers are initialized with linear interpolation kernels and trained on the specific segmentation images. In [7], [8], the authors were able to produce state of the art segmentations by combining the final prediction layer with deconvolutional layers derived from 32x, 16x, and 8x downsampling strides.

The most significant limitation of the fully connected neural networks comes from the loss of resolution accuracy due to the upsampling operation [8]. To understand the limitation, in the updated version of their manuscript [8], the authors suggest a method to deliver upper bounds on performance accuracy. As an approximate upper bound of what can be achieved, the authors took the ground truth segmentation images, downsample them by different factors and then upsample them to the original resolution. They then compare the upsampled images against the unprocessed images to establish an approximation of what can be achieved. Thus, as explored by the creators of the method, it is quite clear that such networks should not be used to detect small regions (such as small coronal holes) or to achieve pixel-level resolution accuracy. For our application, it is also important to note that the size of the coronal holes may appear to be significantly reduced due to missing observations, where the spacecraft was unable to collect any data.

Instead of the FCN approach of combining images from different resolutions, the SegNet architecture considered the development of a simpler auto-encoding architecture [12]. Thus, while both FCN and SegNet performed very well, a fundamental advantage of the SegNet architecture is that is based on learning far fewer parameters. For SegNet, the encoder is still made of convolutional layers followed by max pool layers. However, instead of combining the reduced resolution maps, the decoder is made of just the transposed operations. Thus, the decoder consists of unpool and deconvolution layers that are in the reversed-order (compared to the encoder order). More specifically, the SegNet architecture consists of 4 layers with 64 features with 7x7 convolution kernels, ReLU activation functions, followed by max pool layers over 2x2 non-overlapping regions. For the decoder, the max-sample locations are used to reverse each layer, starting with the last layer and continuing to the first layers, without the use of any activation functions. The final layer consists of SoftMax classification into different categories. A Bayesian extension of SegNet appeared in [13]. In [13], the authors made it very clear that there is strong model uncertainty at the boundaries. Hence, as we shall demonstrate, post-processing SegNet results by a carefully designed level-set method can yield accurate boundary segmentation.

For the the U-net architecture, the authors followed a hybrid approach [14]. As for SegNet, the decoder includes the transpose operations. However, similar to FCN, the decoder combines the decoded outputs with the corresponding encoder outputs from each one of the convolution layers at both the original and reduced-resolution layers. For U-net, the number of parameters to be learned is much larger than for SegNet, but it is also significantly less than for FCN. As for SegNet and FCN, U-net cannot perform accurate boundary segmentation.

As mentioned earlier, we propose a new segmentation method that uses the best-performing methods to initialize a level-set segmentation method. The basic idea is to initialize level-set segmentation close to the coronal-hole boundaries, and then allow level-sets to evolve the coronal-hole boundary curves to the magnetic boundaries. To accomplish this, the level-set function is modified to take into account both the magnetic boundary as well as the local gradient of the EUV image. Furthermore, to reduce artifacts, we post-process the initial segmentation maps to remove invalid coronal holes that may not be unipolar or appear dark.

Based on the segmented images, our goal is to determine the physical map models that best reflect the observations. Then, as mentioned earlier, space weather forecasting is based on the use of physical maps to predict disruptions to the earth's magnetic field. To accomplish this task, we introduce the coronal hole matching problem as an intermediate step.

We demonstrate the coronal hole matching problem in Fig. 2. Each example in Fig. 2 represents a *consensus map* (solid red and blue regions) and the physical model map (hollow and light red and blue regions). Coronal hole matching poses several challenges. First, we have wrap-around effects since the coronal holes are located on a spherical surface (see Fig. 2(a) and 2(b)). Second, coronal hole areas and distances also require geometric corrections. Third, matching needs to consider missing pixel observations (see black pixels in Fig. 1), the generation of new coronal holes by the physical models (see Fig. 2(f)), or the fact that some of the coronal holes may be missing entirely from the physical models (see Fig. 2(g)). Fourth, due to randomization of parameters in the physical model and the fact that observations may come from different times, there is strong variability in the number, location, and the areas of the coronal holes. Thus, instead of matching individual coronal holes, we decided to cluster them together and match their clusters (see Fig. 2).

We summarize the primary contributions of the paper into:

- Development of a new dataset segmentation problem based on consensus maps derived from two independent experts.
- A new segmentation method for detecting coronal holes based on a level-set method initialized by a combination of the Henney-Harvey algorithm [15], SegNet [12], and FCN [7], [8]. The proposed approach achieves substantial improvements over prior methods.
- Development of a new approach for detecting matches, missing, and new coronal holes. Due to the significant physical constraints, our approach is significantly different from any prior method. For example, in [16]–[18], the authors studied a similar problem in cells and nuclei matching. These approaches used heuristics based on shape, size, and empirical set distances. In our case, we develop a Linear Programming model using spherical geometry to compute physical distances and areas derived from their projections. Furthermore, unlike cells, we have the new problems of having to detect the random creation and disappearance of coronal holes.
- Development of a physical map classification system. This new system is built upon the coronal hole segmentation and matching approaches to determine the physical maps that will be used for forecasting.

CR1. Cluster polar coronal holes: Coronal holes that are within 30° from north and south poles are clustered together. Furthermore, any coronal hole that crosses the 30° lines is clustered into the corresponding polar coronal hole. The resulting clusters represent the north and south polar coronal hole clusters.

CR2. Nearby clustering: Coronal holes that are extremely close to each other are clustered together.

CR3. Small-small clustering: Groups of small coronal holes that are relatively close to each other are clustered together.

CR4. Large-small clustering: A small coronal hole that is close to a much larger one is considered part of the larger cluster that involves the larger coronal hole.

CR5. No large-large clustering: In general, larger coronal holes are not clustered together unless they are extremely close to each other (see **CR2**).

Fig. 3: Coronal hole clustering protocol. The manual protocol was used for implementing physically meaningful and consistent rules. The protocol was applied several times to ensure reproducibility and little to no intra-rater variability.

The rest of the paper is organized into four sections. Section II describes carefully developed manual protocols that help in creating consistent ground truth for segmentation, matching and classification. Following this we have Section III, giving a brief introduction to segmentation and a detailed description of matching and classification algorithms. In Section IV we go over results followed by conclusion in Section V.

II. Solar Image Analysis Problem Setup

A. Coronal hole segmentation problem setup

To generate the consensus map, we followed a three-step process. First, a human was trained to manually outline the individual coronal holes based on physical constraints (e.g., unipolarity and appearance, see Fig. 1). Second, the process was repeated with a second human reader. Third, the two human readers got together to generate the consensus maps by agreeing on what constitutes a coronal hole based on a second look and by reviewing their original maps.

B. Coronal hole matching problem setup

In this section, we provide a summary of the process of creating ground truth for the coronal hole matching algorithm. We summarize the manual protocol for clustering coronal holes in Fig. 3 and demonstrate the application of the protocol in Fig. 4.

We describe a manual protocol that ensures reproducibility of the coronal hole matching problem in Fig. 5. The protocol is used to match clusters of the same polarity. The remaining clusters that could not be matched, depending on their presence in the consensus maps, they are labeld as new or missing (e.g., see Fig. 2).

C. Coronal hole map classification problem setup

To ensure reproducibility of the process, we need to take into account that many physical models appear very similar to each other. Thus, to standardize our approach, we group together maps that are virtually indistinguishable and classify each group as opposed to classifying individual maps.

Maps are pre-classified into two groups. We use ranks to describe each group. In the rank 1 group, we include maps that tend to be closer to the consensus map. In the rank 2 group, we include maps that tend to be further away from the consensus map. We then make the final classifications of what constitutes a good and a bad map based on the rules given in Fig. 6.

To decide the rankings, we examine the mid-latitude coronal holes. Initially, similar to clustering, we group maps based on how similar they are to each other. The collection of all of the groups are then classified as being closer to the consensus map (rank 1) or further from the consensus map (rank 2). Here, we classify a group as being closer to the consensus map if it contains a substantial number of matched, fewer cases of new (generated) and missing (removed) coronal holes. A ranked group of maps (rank 1 or 2) is then classified a *good match* if it is in good agreement of the consensus map, where we also allow slight over-estimation of the area of the coronal holes. A group of maps that is not considered a *good match* is classified as a *bad match*. We present a classification example in Fig. 7.

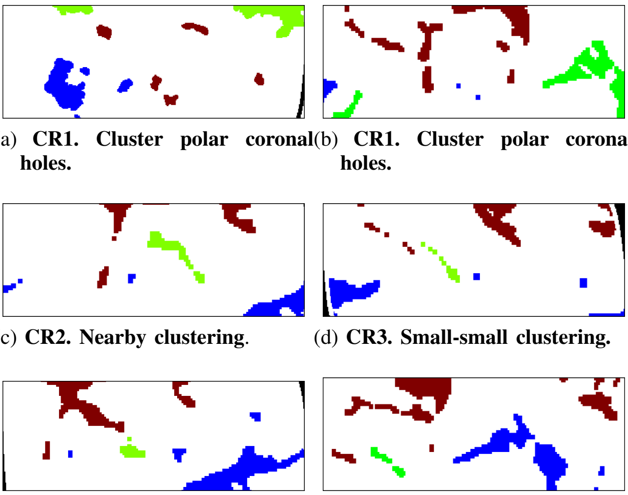

(a) **CR1. Cluster polar coronal holes.**
(b) **CR1. Cluster polar coronal holes.**
(c) **CR2. Nearby clustering**.
(d) **CR3. Small-small clustering.**
(e) **CR4. Large-small clustering.**
(f) **CR5. No large-large clustering.**

Fig. 4: Manual application of coronal holes clustering rules that meet physical requirements. The green coronal holes are combined into a single cluster. Here, CR1 to CR5 refer to the rules that are summarized in Fig. 3. We use red to depict coronal holes with positive polarity. We use blue to depict coronal holes with negative polarity.

M1. Polar to polar matching: Polar clusters with a relatively large area overlap (70% to 100%) are matched.

M2. Polar to mid-latitude matching: A coronal hole cluster from the consensus map that is located in the mid-latitude region is matched to a polar cluster from the physical model when they overlap by at-least 15% to 20%, or more.

M3. Mid-latitude to mid-latitude matching: Mid-latitude clusters are matched with good area overlap (e.g., overlap area $> 30\%$) or weaker area overlap but good localization.

Fig. 5: Cluster matching protocol. The manual protocol was used to produce a reference matching approach for training and testing matching rules. The protocol was applied several times to ensure reproducibility and little to no intra-rater variability.

We show examples from two groups in Fig. 7. Group A maps do not have a matching for the positive polarity coronal hole located in the upper-right region (depicted as bright red). Group B maps do have a matching cluster for the same coronal hole (depicted as faded red). Furthermore, group 2 maps missed (removed) fewer coronal holes (depicted as solid blue here). Thus, group B maps are thus classified as rank 1 and group A maps are classified as rank 2. Furthermore, since rank 1 maps detected more of the coronal holes, rank 1 maps were classified as *good matchings*. On the other hand, since rank 2 maps missed some of the coronal holes, rank 2 maps were classified as *bad matchings*.

III. Methodology

We provide an overview of the proposed approach in Fig. 8. We decompose the proposed system into three components: (i) coronal hole segmentation, (ii) clustering and matching, and (iii) classification. We describe the image segmentation

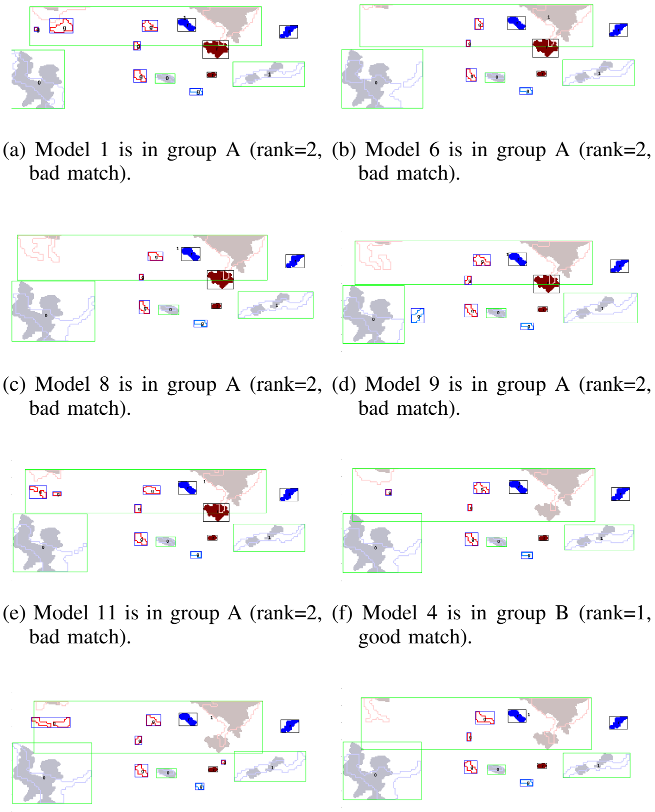
```
Group maps based on matched clusters
Rank groups as rank 1 (better) or rank 2 (worse).

if (both ranks represent good matches) then
    Classify all maps of the day as good matches.
end if

if (both ranks represent bad matches) then
    Classify all maps of the day as bad matches.
end if

if (rank 1 is acceptable and not rank 2) then
    Classify maps of rank 1 as good matches.
    Classify maps of rank 2 as bad matches.
end if
```

Fig. 6: Map classification protocol to ensure consistency. For consistent classification, physical maps are grouped together based on their overall matching performance. Each group is then ranked. Classification is based on the group rank (good or bad).



(a) Model 1 is in group A (rank=2, bad match).

(b) Model 6 is in group A (rank=2, bad match).

(c) Model 8 is in group A (rank=2, bad match).

(d) Model 9 is in group A (rank=2, bad match).

(e) Model 11 is in group A (rank=2, bad match).

(f) Model 4 is in group B (rank=1, good match).

(g) Model 10 is in group B (rank=1, good match).

(h) Model 12 is in group B (rank=1, good match).

Fig. 7: Map classification example (see Fig. 6). The maps are grouped into groups A and B. Group A is classified as rank 2. Group B is classified as rank 1. Maps with rank 1 are classified as good. Maps with rank 2 are classified as bad.

approach in subsection III-A. For matching, we consider each physical map against the segmented map. To reduce variability, we form clusters of coronal holes and attempt to match clusters between maps. The basic method is given in subsection III-B. Physical maps are then classified based on their matching performanced as described in section III-C.

TABLE I: SegNet architecture. The total number of parameters to be learned is 0.52M. The input is the EUV image and the output is a segmentation into three categories (no-observation, coronal hole, other). Here, Conv-bn-ReLU refers to the combination of a convolution layer, batch normalization, and an ReLU activation function.

Layer(s)	Description
Enc1	64 feature maps of 4 3x3 conv-bn-ReLU
Pool1	2x2 Max pooling with stride of [2, 2] (2× downs.).
Enc2	Same as Enc1
Pool2	Same as Pool1 (4× downs. of original)
Unpool1	2x2 Max unpooling (restore at 2× orig. res.)
Dec1	4 3x3 conv-bn-ReLU Layers
Unpool2	Same as Unpool1 (restore original res.)
Dec2	Same as Dec1
SoftMax	Pixel-level classification

TABLE II: FCN architecture. The total number of parameters to be learned is 134.2M. The input is the EUV image and the output is the segmented image.

Layer(s)	Description
Conv1	64 feature maps with two 3x3 conv-ReLU
Pool1	2x2 Max pooling with stride of [2, 2] (2× downs.)
Conv2	128 feature maps with two 3x3 conv-ReLU
Pool2	Same as Pool1 (4× downs. of original)
Conv3	256 feature maps with three 3x3 conv-ReLU
Pool3	Same as Pool1 (8× downs. of original)
Conv4	512 feature maps with three 3x3 conv-ReLU
Pool4	Same as Pool1 (16× downs. of original)
Conv5	Same as Conv4
Pool5	Same as Pool1 (32× downs. of original)
FC6	4096 7x7 convs for 512 features
D6	50% dropout
FC7	4096 fully-connected layer
D7	Same as D6
Fuse1	Combine 16× and 32× downsampled outputs
Fuse2	Combine 8×, 16×, 32× downsampled outputs
SoftMax	Pixel-level classification

A. Multimodal Segmentation of Coronal Holes

We develop a multi-modal segmentation approach as shown in Fig. 8. The basic idea is to provide an initial segmentation map that is then input to a level-set method that is designed to provide an accurate segmentation by evolving the initial segmentation to the magnetic boundary.

To initialize the segmentation approach, we use three different methods working with three different sets of pixel resolutions. At the pixel resolution level, we use the original Henney-Harvey method [15] that uses pixel classification to select candidate coronal holes that are both unipolar and darker in the EUV images. At the 2× and 4× downsampled resolution levels, SegNet reconstructs pixel-level segmentations by first encoding the EUV image at 1/2 and 1/4 of the original image resolution and then decoding the encoded maps to provide classification at the full resolution. At the 2× to 32× downsampled resolution levels, FCN reconstructs pixel-level segmentations by first encoding at 1/2 to 1/32 of the original image resolution, and then combining $8\times, 16\times, 32\times$ encodings to form the predicted segmentation image at the full resolution.

For the neural-net based methods, we only input the EUV images since they are used for visual classification by the astronomers and avoid combining them with the magnetic images. For the magnetic images, note that the training set is rather limited, there is no possibility of pre-training since they are very different from visual images used in standard datasets, and we defer further processing by the coronal-hole selector and level-set methods. Here, we also note that we had to learn 134.2M parameters for FCN (pre-trained on VGG-16 [10], [19]) and 0.52M parameters for SegNet. In any case, we also verified these claims experimentally. As we describe in the results section, the performance of SegNet when input with both the magnetic and EUV images deteriorated considerably while there was no observed improvement for FCN.

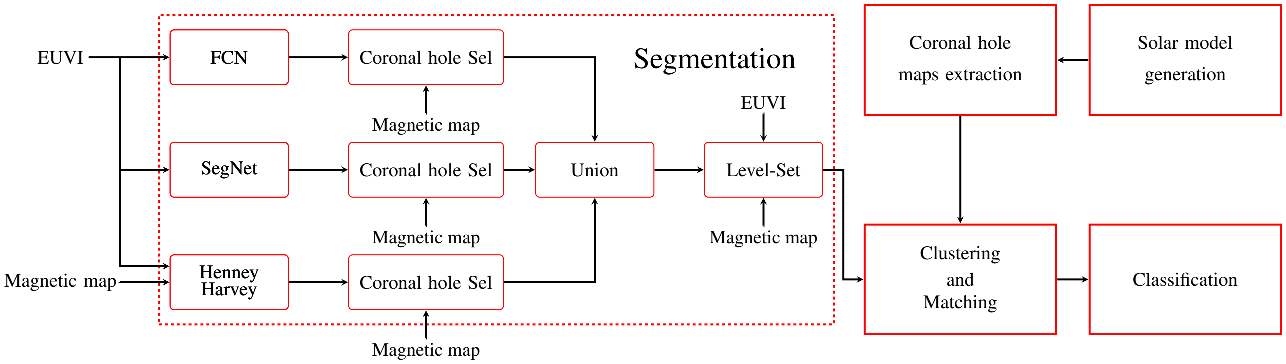


Fig. 8: General system diagram. A collection of coronal hole maps are generated for different physical model parameters. The input observations are used to generate a candidate coronal hole map. A hierarichical clustering and matching algorithm is used for matching clusters of coronal holes. The final classifier is based on the matching results.

We provide more details on the specific SegNet and FCN architectures in Tables I and II. All of the parameters were learned on the training set. The training set represents a random sample of 70% of the images. None of these images overlap with the remaining test set that are used in the Results section. For both methods, we used 50 epochs with a mini-batch size of 7. For SegNet, we used a learning rate of 0.1 with a momentum set to 0.9. It took about an hour to train the SegNet on dual NVIDIA GTX 1080 video cards with 2,560 cores and 8GB each. On the same system, it took about 20 minutes for the pre-trained FCN to be re-trained using a learning rate of 10^{-3} and a momentum parameter set to 0.9.

The coronal holes resulting from the initial segmentations are classified (selected) to further impose physical constraints on their appearance. Thus, since coronal holes are supposed to be dark in EUV images and unipolar in the magnetic images, we use histograms of EUV intensity (255 bins) and magnetic flux (40 bins) as input features to the classifier. Furthermore, we add the area of each coronal hole as an input feature since the downsampling and upsampling operations generate small, noisy estimates of the actual coronal holes. For classifying each coronal hole, we use a Random Forest for each segmentation method.

To select the parameters for the Random Forest, we further split the training set into 70% training for each classifier model and 30% testing for selecting the best classifier model ([20]). We used the out-of-bag error determine the best models that avoid over-fitting: (i) 50 trees with 50 splits for Henney-Harvey, (ii) 30 trees with 50 splits for SegNet, and (iii) 30 trees with 20 splits for FCN.

After identifying candidate coronal holes, we use a union operation to combine the outputs. Here, we recognize that the union operation will likely overestimate the actual coronal holes. However, given the potentially catastrophic consequences of missing an actual coronal hole, we prefer to err on the side of providing a slight over-estimation as opposed to missing one of them. The union map is then input to the level-set method that can provide an accurate estimation of the magnetic boundary.

1) Level-set Segmentation: We develop a multi-modal segmentation approach by expanding the Distance Regularized Level Set Evolution (DRLSE) method [21] to account for physical constraints on the Extreme Ultra Violet (EUV) images and photo maps. Our physical constraints include: (i) coronal holes appear darker in EUV images [22], (ii) they are unipolar in photo map images (positive or negative values only), and (iii) they are not allowed to cross magnetic neutral lines (zero crosses in photo map images) [23].

```
function LS(EUVI, mag_img, init_img)
    ▷ Input: EUVI and mag images to process.
    ▷ Output: Segmented image.

    $I$ ← smooth EUVI with 15 × 15 Gaussian kernel
              with optimization variable $\sigma$.
    g ← $\frac{1}{1+(I_x^2+I_y^2)}$

    ▷ Make g zero at magnetic boundaries
    p  ← DetectMagneticCrossLines(mag_img)
    pg ← (1 - p ) .* g

    ▷ Init. using combination of Henney-Harvey [15],
    ▷ SegNet [12], and FCN [7].
    $\phi$ ← init_img
    for i $\leq$ n do
        $\delta(\phi)$ ← Dirac($\phi$, $\epsilon$)

         ▷ Use modified edge function pg:
        $F_a$ ← areaTerm($\delta(\phi)$, pg)
        $F_e$ ← edgeTerm($\delta(\phi)$, $\phi$, pg)
        $F_d$ ← Regularize_distance($\phi$)

         ▷ Allow $\alpha$ to vary for optimization:
        $\phi$ ← $\phi$ + ts·($\mu F_d$ + $\lambda F_e$ + $\alpha F_a$)
    end for
return $\delta(\phi)$
end function
```

Fig. 9: Level-set segmentation algorithm using the modified edge function pg.

To summarize the level-set segmentation method, we review the basic definitions given in [21]. Let $p(.)$ be used for defining a regularized distance for the level set function (ϕ), $g(.)$ denote the edge function that should be minimized at image edges, $G(.)$ denote the Gaussian, and ∇ denote the gradient operator. We then define the divergence operator using $d_p(s) \triangleq p'(s)/s$, the edge function $g = 1/(1+|\nabla G * I|^2)$, and a localization function $\delta_\epsilon(x)$ that is zero for $|x| > \epsilon$ and non-zero for $|x| < \epsilon$. We modify the edge function so that it does not allow crossing the magnetic neutral lines. This is accomplished by modifying the edge function to be:

$$pg = (1-p)g \tag{1}$$

where p assumes the value of 1 over the magnetic polarity boundaries detected in the magnetic image and is zero away from the boundary. Thus, over the magnetic lines, the edge function becomes zero and prevents crossing of the neutral line boundary.

The segmented image is computed by evolving the level set as given by:

$$\frac{\partial \phi}{\partial t} = \mu \mathcal{R}_p(\phi) + \lambda \mathcal{L}_{pg}(\phi) + \alpha \mathcal{A}_{pg}(\phi) \tag{2}$$

where:

$\mathcal{R}_p(\phi) = \mathrm{div}(\mathrm{d_p}(|\nabla\phi|)\nabla\phi)$ is the distance term,

$\mathcal{L}_{pg}(\phi) = \delta_\epsilon(\phi) \cdot \mathrm{div}\left(pg \cdot \frac{\nabla\phi}{|\nabla\phi|}\right)$ is the boundary term, and

$\mathcal{A}_{pg}(\phi) = pg \cdot \delta_\epsilon(\phi)$ is an area term.

We provide a description of the proposed level-set segmentation algorithm in Fig. 9. The approach requires joint processing of the EUV and magnetic images. From (2), we have found that α and the spatial spread of the Gaussian (σ) used for computing the edge function are the two parameters that can affect overall segmentation performance. To find the optimal parameter values, we compare against the consensus maps, and look for the optimal values:

$$\min_{\alpha,\,\sigma} \sqrt{[1-\texttt{spec}(\alpha,\sigma)]^2 + [1-\texttt{sens}(\alpha,\sigma)]^2} \tag{3}$$

where Spec denotes the (pixel-level) specificity and Sens denotes the corresponding sensitivity. The solution of (3) gives the optimal values for each image. For each image, we constrain the optimization problem for $\alpha \in [-3, +3]$, $\sigma \in [0.2, 1]$. Over the training set, we select the median values over the entire set.

The optimization of (3) is challenging since derivative estimates can be very noisy. To this end, we use a robust optimization method based on Pattern-search initialized with $\alpha_0 = 0, \sigma_0 = 0.5$. We refer to [24] for details on the optimization procedure.

B. Clustering and Matching

We use the segmented maps as reference maps for comparing against the physical models as detailed in Fig. 10. For each date, we compare each physical model against the reference map to determine new, missing, and matching coronal hole clusters. In what follows, we provide more details for each step.

```
function CLUSTER_MATCHING(dates)
▷ Input:   dates to process. Each date has
           an associated list of physical models and a
           a reference (segmentation) map.
▷ Output:  model_maps of the new, missing, and
           matched coronal holes for each physical model
           and date are stored as separate files.

    ▷ Process each date separately
    for date ∈ dates do
        ▷ Read and process reference image
        ref_map ← load_ref_data(date)
        $ref\_map_{\{+,-\}}$ ← pre_process(ref_map)

        ▷ Process associated physical models
        for model ∈ {model_1, ..., model_M} do
            model_map ← load_model(date, model)
            $model\_map_{\{+,-\}}$
                    ← pre_process(model_map)

            ▷ Analyze each polarity separately
            for polarity p ∈ { +, −} do
                Cluster coronal holes that are are close.
                Detect coronal hole clusters that are in
                    physical maps but not in reference
                    map using Mahalanobis distance
                    and store the results in
                    $new\_map_p$ and $missing\_map_p$
                Re-cluster remaining coronal holes in
                    ref_map and model_map to
                    achieve equal number of clusters.
                Match clusters using linear programming
                    and save the results in
                    $matched\_map_p$
            end for

            ▷ Save maps for each physical model
            model_maps ← ($new\_map_p$,
                $missing\_map_p$, $matched\_map_p$)
                for polarity p ∈ {+, -}.
            Store model_maps
        end for
    end for
end function
```

Fig. 10: Cluster matching algorithm. The algorithm process the segmentation map (reference map) against the physical models for each given date. The resulting maps contain the new, missing, and matched coronal holes between the reference map and each physical model. In our example, we have 12 physical maps ($M = 12$).

1) Pre-processing: Pre-processing is summarized in Fig. 11. Each physical map is resized from 144×172 to 360×180 using bilinear interpolation to match the resolution of the EUV, magnetic, and final segmentation maps. Then, we remove the polar regions and regions where we have no observations. We

```
function PRE_PROCESS(map)
▷ Input:   map to process.
▷ Output: $\texttt{map}_{\{+,-\}}$ contains separate coronal hole
          for positively and negatively charged coronal
          holes of the same size as the magnetic images,
          with polar regions removed.

    ▷ Extract reference and model maps
    map ← extract_coronal_hole_maps (map)

    ▷ Remove small gaps in the maps
    map ← binary_close (map)

    ▷ Resize maps to photomap image size
    map ← resize (ref_map)

    ▷ Separate the positive from the negatively charged
       coronal holes
    $\texttt{map}_{\{+,-\}}$ ← extract_polarity_maps (map)

    ▷ Remove regions where there are no observations,
      and regions with latitude 0 to 30 degrees, and
      150 to 180 degrees
    $\texttt{map}_{\{+,-\}}$ ← remove_regions ($\texttt{map}_{\{+,-\}}$)

    return $\texttt{map}_{\{+,-\}}$
end function
```

Fig. 11: Pre-processing steps for cluster matching.

return the positive and negative maps separately for further processing.

2) Clustering: The initial clusters are formed from coronal holes that are close to each other. To measure distance, let A and B define the set of pixels that belong to two coronal holes of the same polarity. We define set distance using $d(A, B) = \min_{x \in A, y \in B} d(x, y)$ where $d(x, y)$ is based on distance. Thus, initially, we cluster A and B into the same cluster if $d(A, B) < T$ where T is a threshold value that is determined during training. Here, we note that the relationship is transitive. In other words, if $d(A, B) < T$ and $d(B, C) < T$, we cluster A, B, C into the same cluster.

3) Detecting new and missed coronal hole clusters: After clustering, there is a need to compare the reference map to the model maps to detect significant differences. We use the Mahalanobis distance to determine coronal hole clusters that are sufficiently close for possible matching. For computing the Mahalanobis distance, we compare the physical areas and set distances (minimum physical distances) between coronal hole clusters.

The remaining coronal hole clusters are assumed to be too far apart to be matchable. Then, from the remaining ones, coronal hole clusters that are in the reference map but are missing from the model maps are classified as *missing* and are added to the $\texttt{missing\_map}_p$ (for the p-th model map). Similarly, coronal hole clusters that are in the model map but are missing from the reference map are classified as *new* and are added to the $\texttt{new\_map}_p$ (for the p-th model map).

4) Matching with re-clustering: After removing the new and missing coronal hole clusters, the remaining ones need to be matched. Unfortunately, we can still have different numbers of clusters in each map. Thus, instead of matching maps having different number of clusters, we need to first combine them together to have equal numbers of clusters. Clustering is accomplished using the minimum physical distance between coronal holes. Here, we note that our use of the minimum spherical distance takes into account wrap-around effects. The basic idea is to iteratively cluster together all coronal hole clusters that are separated by a minimal physical distance until we reach the desired number of clusters. We introduce linear programming model for computing an optimal matching between coronal hole clusters.

Let i be used to index clusters in the reference map. Similarly, let j be used to index clusters in the physical map. Then, we use $m_{i,j}$ to denote a possible match between cluster i in the reference map and cluster j in the physical map. Thus, $m_{i,j} = 1$ when there is a match between the clusters and $m_{i,j} = 0$ otherwise. We also assign a cost $w_{i,j}$ associated with the matching. Here, we set the $w_{i,j}$ to be the shortest spherical distance between the clusters (set distance). Thus, $w_{i,j} = 0$ when the clusters overlap.

Formally, we find an optimal matching by solving:

$$\min_{m_{i,j}} \sum_i \sum_j w_{i,j} m_{i,j} \tag{4}$$

subject to:

$$\sum_i m_{i,j} = 1, \tag{5}$$

$$\sum_j m_{i,j} = 1 \tag{6}$$

$$m_{i,j} \in \{0, 1\} \tag{7}$$

where $m_{i,j}$ denotes the assignment that minimizes the weighted matching of (4), while each cluster can only be assigned to one other cluster as required by (5) and (6). This is a typical bipartite matching setup, making matching matrix created from $m_{i,j}$ to be totally unimodular. As discussed in [25] this problem when solved with linear programming will return an integer solution.

C. Classification

After cluster matching, we need to make a decision on whether the physical model is sufficiently good for forecasting applications. Based on the cluster matching results, we extract the following features for classification: (i) number of new coronal hole clusters, (ii) number of missing coronal holes, (iii) total physical area of new coronal hole clusters, (iv) total physical are of missing coronal hole clusters, and (v) overestimate of physical area of coronal holes as estimated by the physical model. Here, we note that the area overestimate comes from the need to avoid underestimating the impact that the coronal holes will have on the earth. The basic idea is that the excessive area will likely account for more potential events that should not be missed by the physical models.

For the final classifier, we use a Random Forest. Here, we note that a random forest classifier produces a majority classification based on a collection of tree classifiers. For each tree, decisions are represented as inequalities implemented on the five basic features that we have just described. The majority vote can significantly reduce the variance of the resulting classifier (see [20], [26] for details).

IV. Results

The results are summarized in five sections. First, we provide a description of the dataset in section IV-A. We then show an example of image segmentation in section IV-B. We summarize results for coronal hole matching in section IV-C, and final classification results in section IV-D.

A. *Dataset*

The dataset consisted of two Carrington rotations [27] for a total of 50 days. The first Carrington rotation covers the dates from 07/13/2010 to 08/09/2010. The second Carrington rotation covers the dates from 01/20/2011 to 02/16/2011. For each day, we used:

- The EUV (synoptic) image (see [28])
- Magnetic photo map image
- Expert segmentations by two independent expers
- The Consensus map derived from the expert segmentations
- Twelve physical model maps manually classified as Good or Bad based on their agreement with the corresponding consensus map.

For each day we have 12 coronal hole forecasts. Hence, for training the random forest classifier, we consider 600 images that correspond to 12 physical models per day (50 days total).

For level-set segmentation, we used 70% of the images for training and 30% for testing. We report segmentation results on the test set. Similarly, for physical map classification, we train the classifier on 70% (419 maps) of the maps and report the results on the remaining 30% (181 maps).

B. *Segmentation*

We demonstrate the performance of the proposed segmentation algorithm in Fig. 12. The inputs are the EUV and the magnetic images appearing in (a) and (b). The ground truth consensus segmentation is shown in (c). Initial segmentations for (d) Henney-Harvey (distance from ($\texttt{Sens}$=1, $\texttt{Spec}$=1) is d=0.36), (e) FCN (d=0.12), and (f) SegNet (d=0.18). In (d), (e), (f), we use green lines to mark the selected valid coronal holes and red to mark coronal holes that were rejected by the random-forest coronal hole selector. In (g), we show the initial segmentation boundary in red and its final evolution in red. In (h), we compare the final segmentation against the consensus maps (d=0.09). The coronal holes with yellow outlines are in agreement with the Consensus map. The coronal holes with pink outlines are not in agreement with the Consensus map. The final segmentation results are also shown on the EUV images in (h). An unoptimized Matlab implementation of Level-set segmentation takes about 38 seconds on a 2.2 GHz Intel Xeon. For real-time level-set implementations for video data, we refer to [29]. For our application, there was no need to improve the speed of the segmentation method.

From the results in Fig. 12, we can see that the Random Classifier selector is very effective. A large number of invalid coronal holes were removed from the initial segmentations of Figs 12(d)-(f) (marked red). On the other hand, we can see that we have also achieved a slight over-segmentation in Fig. 12(h). As discussed earlier, we prefer to over-segmentation over the possibility that we may miss a valid coronal hole. As can be seen from the results in Fig. 12(h), the level-set method propagated the coronal-hole boundary to the local magnetic boundary.

To document the accuracy provided by the level-set method, refer to Table III. From Table III, it is clear that level-set segmentations can be used to provide substantial improvements over all other methods that were considered. Furthermore, from the results in Table III, it is clear that the addition of the magnetic maps did not improve segmentation for FCN and SegNet. For the proposed method, we considered the full-segmentation method described in Fig. 8. We also show results for individual dates in Fig. 13. Overall, the proposed method performed extremely well. In comparison, the error ranges from 2× to 3× for Henney-Harvey (two to three times larger), 1.5× to 2× for FCN, 2× to 3× for SegNet, and 3× to 5× for U-net.

C. *Coronal hole matching*

We present the results from coronal hole matching between the Physical maps and the Consensus maps in Table IV. The results are compared against manual matching each coronal hole. Overall, at 92.8%, the method produced 7,522 matches, in agreement with manual labeling. The proposed approach failed to match 776 coronal holes (776=416+360) while it overmatched 25 (25=22+3) of them. Thus, compared to the human expert, the proposed approach tends to undermatch. Nevertheless, at 92.8%, the proposed method performed very well.

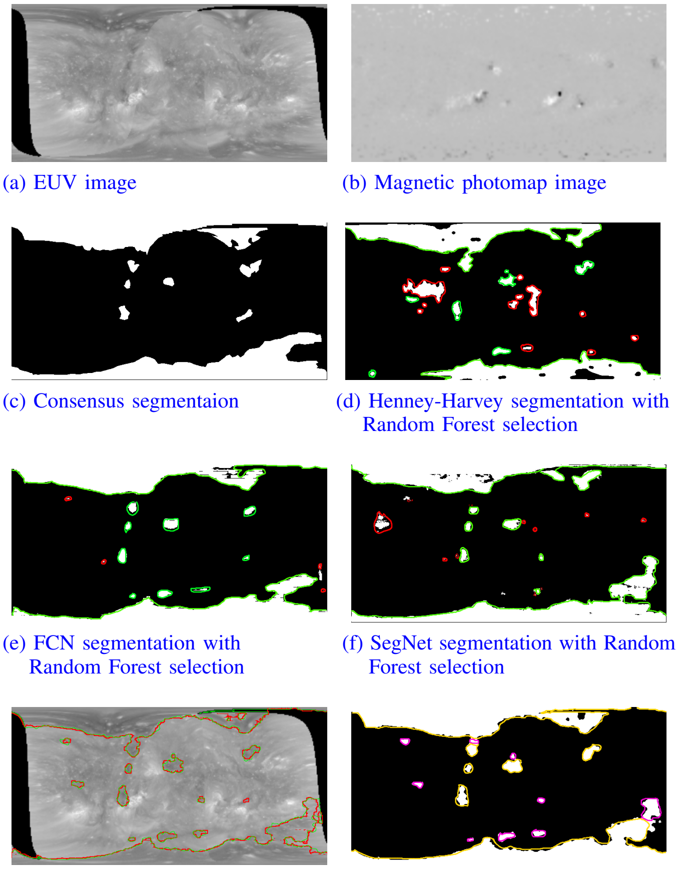
Results of matching using linear programming are demonstrated in Figs. 14 and 15. Fig. 14 shows an example where automated matching agrees with manual matching. Fig. 15 shows an example where there are significant differences between automated and manual matching (see pink cluster).

D. *Classification results*

We present the results from training the Random Forest classifier in Fig. 16. To train the classifier, we further split the training set into a 70% training set and a 30% test set as recommended in [20]. The results over the test set (part of the original training set) were used for determining the optimal parameters. The out-of-bag error (OOB) error is shown as function of the number of trees and the maximum tree depth in Fig. 16. From the graph, we determine that the optimal configuration will be to use 20 decision trees at a maximum depth of 11.

TABLE III: Segmentation results as a function of distance from the ideal result (`Sens`=1, `Spec`=1). Level-set segmentation improved all prior methods. We trained a 3-stage U-net with 7M parameters, a learning rate of 10^{-4} over 1000 epochs. For the rest of the methods, refer to section III-A for how the segmentation methods were trained. We list the ratio of the achieved performance over the proposed method in parenthesis. Thus, 1× refers to the performance of the proposed method. A value of 2× implies that the error was twice as large as the proposed method.

Method	**Distance from** (`sens`, `spec`) = $(1, 1)$				
	min	**25%**	**med**	**75%**	**max**
U-Net (EUVI)	0.29 (4.8×)	0.31 (3.8×)	0.34 (3.7×)	0.36 (3.6×)	0.39 (3×)
Henney-Harvey	0.12 (2.00×)	0.17 (2.12×)	0.22 (2.44×)	0.27 (2.70×)	0.39(3.00×)
Henney-Harvey + Level-Sets	***0.07*** **(1.16×)**	***0.12*** **(1.50×)**	***0.18*** **(2.00×)**	***0.25*** **(2.50×)**	***0.30 (2.30×)***
FCN (EUVI)	0.09 (1.50×)	0.13 (1.62×)	0.15 (1.66×)	0.19 (1.90×)	0.25 (1.92×)
FCN (EUVI + Mag)	0.10 (1.66×)	0.13 (1.62×)	0.15 (1.66×)	0.2 (2.00×)	0.26 (2.00×)
FCN (EUVI) + Level-Sets	***0.06 (1.00×)***	***0.09 (1.12×)***	***0.11 (1.22×)***	***0.13 (1.30×)***	***0.17 (1.30×)***
SegNet (EUVI)	0.14 (2.33×)	0.17 (2.12×)	0.19 (2.11×)	0.24 (2.40×)	0.30 (2.30×)
SegNet (EUVI + Mag)	0.17 (2.83×)	0.23 (2.12×)	0.36 (2.11×)	0.44 (2.40×)	0.58 (2.30×)
SegNet (EUVI) + Level-Sets	***0.11 (1.83×)***	***0.13 (1.62×)***	***0.14 (1.55×)***	***0.16 (1.60×)***	***0.21 (1.61×)***
Full segmentation method	***0.06 (1×)***	***0.08 (1×)***	***0.09 (1×)***	***0.10 (1×)***	***0.13 (1×)***



(a) EUV image

(b) Magnetic photomap image

(c) Consensus segmentaion

(d) Henney-Harvey segmentation with Random Forest selection

(e) FCN segmentation with Random Forest selection

(f) SegNet segmentation with Random Forest selection

(g) Level-sets segmentation overlaid on EUV image

(h) Proposed segmentation method vs Consensus map

Fig. 12: Proposed segmentation method for the test date case of 8-7-2010.

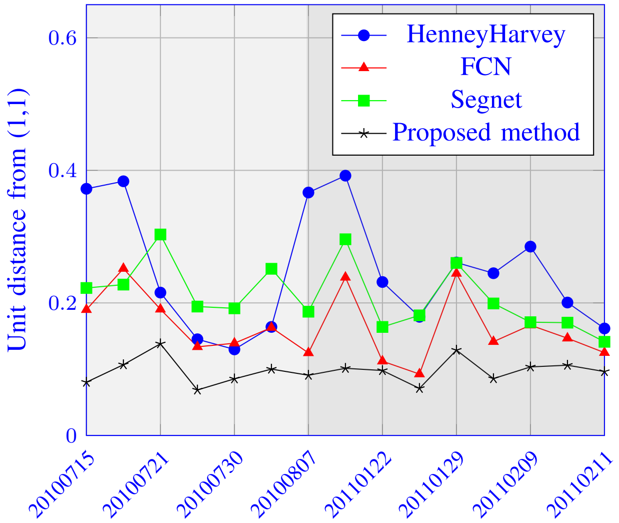


Fig. 13: Comparative segmentation results for different dates. Results over the first cycle are shown on the left. Results over the second cycle are shown on the right (shaded region).

We present classification results over the training set in Table V. Table V summarizes the results over the remaining 30% of the data that were left for testing. Overall, the classifier had a very accuracy at 95.5%.

We also examine the relative importance of each feature in the Random Forest classifier and report the results in Fig. 17.

TABLE IV: Coronal hole matching results. *New* refers to coronal holes that appear in the physical model but are not seen in the consensus segmentation maps. *Missing* refers to coronal holes that appear in the consensus maps but are not produced in the physical model. Note that by definition, a coronal hole cannot be both new and missing. The proposed method achieved an overall accuracy of 92.8%.

Manual	**Proposed Method**			
	New	Missing	Matched	Total
New	**1958**	0	22	1980
Missing	0	**868**	3	871
Matched	416	360	**7522**	8298
Total	2374	1228	7547	11149

Here, we note that the relative importance is a measure of the predictive performance of each variable [20]. From Fig. 17, the most important features come from the area features. Most importantly, a bad physical map will be the result of large amounts of missing area. Similarly, a good physical map will share the same area or have slightly more area.

V. Conclusion

The manuscript summarizes the development of several new solar image analysis methods: (i) Coronal hole detection method, (ii) new segmentation method to detect coronal holes, (iii) manual protocol to support reproducible classification of physical models, (iv) coronal hole clustering and

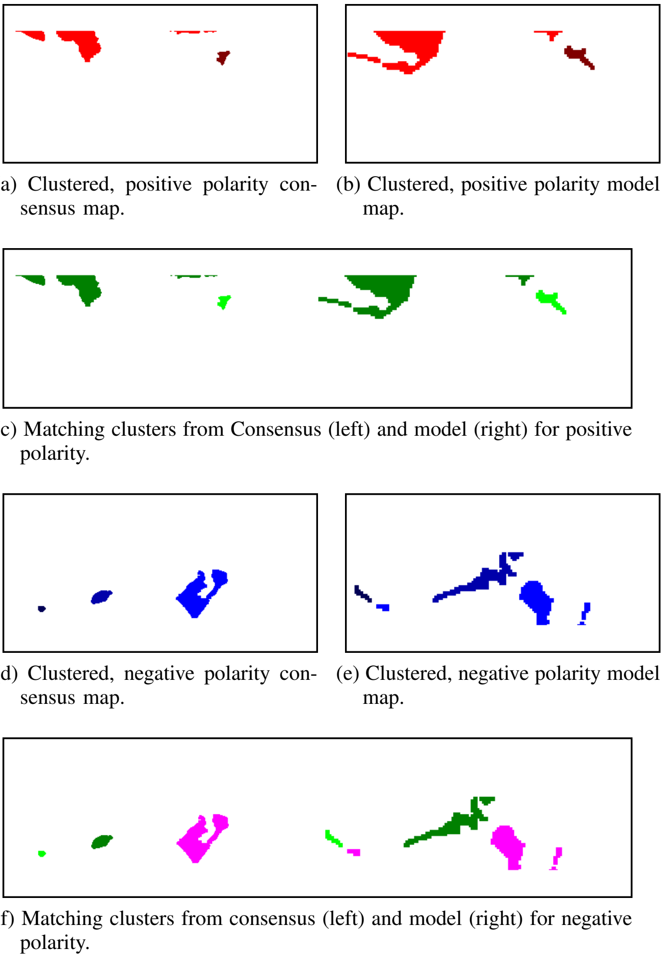

(a) Clustered, positive polarity consensus map.
(b) Clustered, positive polarity model map.
(c) Matching clusters from Consensus (left) and model (right) for positive polarity.
(d) Clustered, negative polarity consensus map.
(e) Clustered, negative polarity model map.
(f) Matching clusters from consensus (left) and model (right) for negative polarity.

Fig. 14: Coronal hole cluster matching example (02-04-2011). Matched clusters share the same color.

TABLE V: Classification results for selecting physical maps for forecasting. We classify a physical map as *good* if it should be used for forecasting. On the other hand, a *bad* physical map is not appropriate for forecasting. Overall classification accuracy is 95.5%.

Manual	**Proposed Method**		
	Bad	Good	Total
Bad	**76**	3	79
Good	5	**97**	102
Total	81	100	181

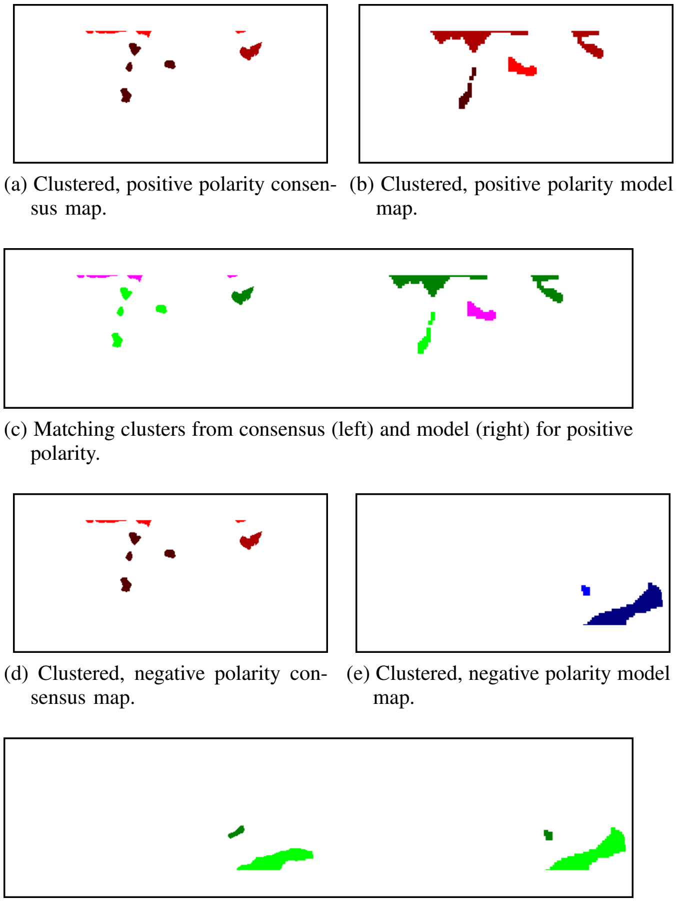

(a) Clustered, positive polarity consensus map.
(b) Clustered, positive polarity model map.
(c) Matching clusters from consensus (left) and model (right) for positive polarity.
(d) Clustered, negative polarity consensus map.
(e) Clustered, negative polarity model map.
(f) Matching clusters from consensus (left) and model (right) for negative polarity.

Fig. 15: Difficult example of coronal hole matching showing issues in the algorithm (07-08-2010). Matched clusters share the same color.

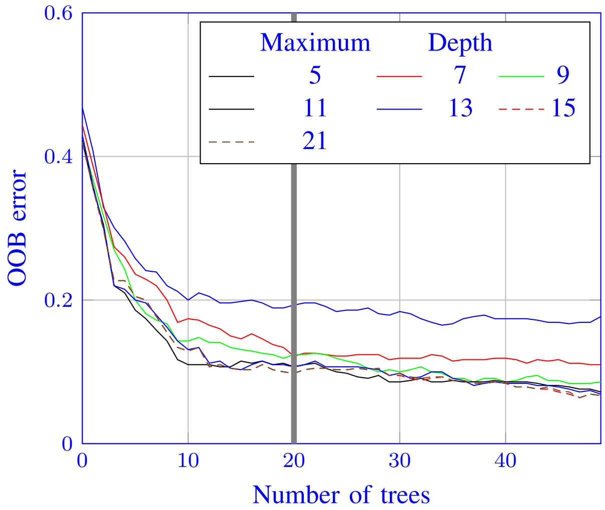


Fig. 16: Out-of-bag (OOB) error used to determine the optimal parameters for the Random Forest.

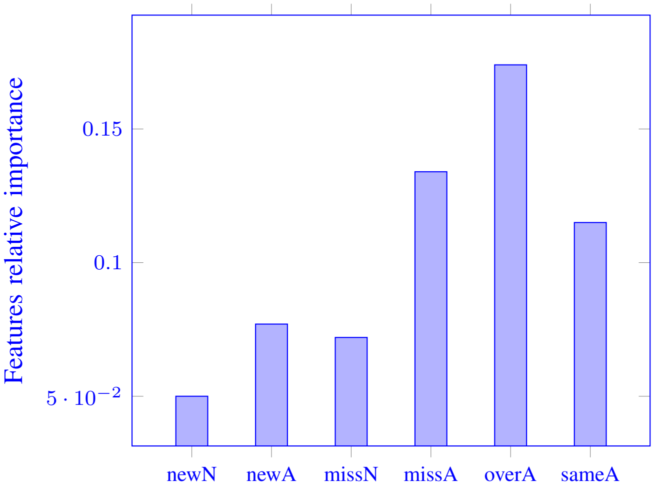


Fig. 17: Relative importance of features from random forest. The features from left to right are `newN`=number of new coronal holes, `newA`=image area of new coronal holes, `missN`=number of missing coronal holes, `missA`=image area of missing coronal holes, `overA`=spherical area overestimated by model, `sameA`=spherical area overlap.

matching system using, and (v) a fully automated method for physical model classification. The performance of each method is validated against human experts. The proposed segmentation method significantly outperformed several other segmentation methods. The code and datasets are available at https://github.com/venkatesh369/HeliosTransactions.

The significance of the proposed method is that it can support forecasting based on space weather models. Nevertheless, prior to deployment, we will need to perform large-scale studies.

Acknowledgments

The authors would like to acknowledge Dr. R. Hock and Callie Darsey for contributing to the manual protocols and helping with the manual segmentation.

References

[1] R. H. Munro and G. L. Withbroe, "Properties of a coronal" hole" derived from extreme-ultraviolet observations," *The Astrophysical Journal*, vol. 176, p. 511, 1972.

[2] C. Arge, J. Luhmann, D. Odstrcil, C. Schrijver, and Y. Li, "Stream structure and coronal sources of the solar wind during the may 12th, 1997 cme," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 66, no. 15, pp. 1295–1309, 2004.

[3] J. A. Linker, Z. Mikić, D. A. Biesecker, R. J. Forsyth, S. E. Gibson, A. J. Lazarus, A. Lecinski, P. Riley, A. Szabo, and B. J. Thompson, "Magnetohydrodynamic modeling of the solar corona during whole sun month," *Journal of Geophysical Research: Space Physics*, vol. 104, no. A5, pp. 9809–9830, 1999.

[4] C. N. Arge and V. J. Pizzo, "Improvement in the prediction of solar wind conditions using near-real time solar magnetic field updates," *Journal of Geophysical Research: Space Physics*, vol. 105, no. A5, pp. 10 465–10 479, 2000. [Online]. Available: http://dx.doi.org/10.1029/1999JA000262

[5] C. N. Arge, D. Odstrcil, V. J. Pizzo, and L. R. Mayer, "Improved method for specifying solar wind speed near the sun," in *AIP Conference Proceedings*, vol. 679, no. 1. AIP, 2003, pp. 190–193.

[6] C. Arge, J. Luhmann, D. Odstrcil, C. Schrijver, and Y. Li, "Stream structure and coronal sources of the solar wind during the may 12th, 1997 cme," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 66, no. 15, pp. 1295–1309, 2004.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[8] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv preprint arXiv:1605.06211*, 2016.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[12] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[13] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[15] C. Henney and J. Harvey, "Automated coronal hole detection using he i 1083 nm spectroheliograms and photospheric magnetograms," in *ASP Conf.*, vol. 346, 2005, pp. 261–268.

[16] X. Chen, X. Zhou, and S. T. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 4, pp. 762–766, 2006.

[17] F. Li, X. Zhou, J. Ma, and S. T. Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *IEEE transactions on medical imaging*, vol. 29, no. 1, pp. 96–105, 2010.

[18] M. A. A. Dewan, M. O. Ahmad, and M. Swamy, "Tracking biological cells in time-lapse microscopy: an adaptive technique combining motion and topological features," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1637–1647, 2011.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, 2nd ed. Springer series in statistics New York, 2009, vol. 1.

[21] C. Li, C. Xu, C. Gui, and M. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, Dec 2010.

[22] M. D. Altschuler, D. E. Trotter, and F. Q. Orrall, "Coronal holes," *Solar Physics*, vol. 26, no. 2, pp. 354–365, 1972.

[23] S. Antiochos, C. DeVore, J. Karpen, and Z. Mikić, "Structure and dynamics of the sun's open magnetic field," *The Astrophysical Journal*, vol. 671, no. 1, p. 936, 2007.

[24] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.

[25] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1982.

[26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[27] R. Howard and B. J. Labonte, "Surface magnetic fields during the solar activity cycle," *Solar Physics*, vol. 74, no. 1, pp. 131–145, 1981.

[28] M. D. Altschuler, R. H. Levine, M. Stix, and J. Harvey, "High resolution mapping of the magnetic field of the solar corona," *Solar Physics*, vol. 51, no. 2, pp. 345–375, 1977.

[29] Y. Shi and W. C. Karl, "A fast level set method without solving pdes [image segmentation applications]," in *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 2. IEEE, 2005, pp. ii–97.