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 approach consists of selecting the physical models based on their agreement with coronal holes extracted from the images.

Ultimately, the goal is to use physical models to predict 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. 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 by providing accurate boundary detection. Overall, the method gave a 95.5% map 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.

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 segmentations by a single human expert can produce significant inter-
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 (e.g., from AlexNet [9], VGG net [10], and GoogLeNet [11]) and add multi-scale deconvolutional layers to predict ground truth segmentation maps from the 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 upsampled them to the original resolution. They then compared 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 it 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 feature maps with 7x7 convolution kernels, ReLU activation functions, followed by max pool layers. However, instead of combining the final 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 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 FCN. As for SegNet and FCN, U-net cannot perform accurate boundary segmentation.

As mentioned earlier, we propose a new approach that uses the best-performing algorithms 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 light red 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, 6]. The proposed approach achieves substantial improvements over prior methods.
- Development of a new approach for matching, and detecting missing or the appearance of 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

**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 conclusions 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 reader 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, are labeled as new or missing (e.g., see Fig. 3).

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 with 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.

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 protocol:

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

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

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

9: if (rank 1 is acceptable and not rank 2) then
10: Classify maps of rank 1 as good matches.
11: Classify maps of rank 2 as bad matches.
12: 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).
TABLE I: SegNet 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 |
|----------|-------------|
| 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 |
| Poo1     | Same as Poo1 (4× downs. of original) |
| Dec1     | 4 3x3 conv-bn-ReLU Layers |
| Unpool1  | 2x2 Max unpooling (restore at 2× orig. res.) |
| 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 Conv1 (4× downs. of original) |
| Conv3    | 256 feature maps with three 3x3 conv-ReLU |
| Pool3    | Same as Conv1 (8× downs. of original) |
| Conv4    | 512 feature maps with three 3x3 conv-ReLU |
| Pool4    | Same as Conv1 (16× downs. of original) |
| Conv5    | Same as Conv4 |
| Pool5    | Same as Conv1 (32× downs. of original) |
| FC6      | 4096 7x7 convs for 512 features |
| FC7      | 4096 fully-connected layer |
| D6       | 50% dropout |
| 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 pixels to classify 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 the original image resolution, and then combining 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.
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, which 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 to 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].

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 \((\phi), g(\cdot)\) denote the edge function that should be minimized at image edges, \(G(\cdot)\) denote the Gaussian, and \(\nabla\) 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:

\[
p_g = (1-p)g
\]

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:

\[
\partial \phi \over \partial t = \mu \mathcal{R}_p(\phi) + \lambda \mathcal{L}_p(\phi) + \alpha \mathcal{A}_p(\phi)
\]

where:

\[
\mathcal{R}_p(\phi) = \text{div}(d_p(\nabla \phi) \nabla \phi)
\]

is the distance term,

\[
\mathcal{L}_p(\phi) = \delta_\epsilon(\phi) \cdot \text{div} \left( p_g \frac{\nabla \phi}{|\nabla \phi|} \right)
\]

is the boundary term, and

\[
\mathcal{A}_p(\phi) = p_g \cdot \delta_\epsilon(\phi)
\]

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 \(\alpha\) and the spatial spread of the Gaussian \((\sigma)\) 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:

\[
\min_{\alpha, \sigma} \sqrt{\left[1 - \text{Spec}(\alpha, \sigma)\right]^2 + \left[1 - \text{Sens}(\alpha, \sigma)\right]^2}
\]

where \text{Spec} denotes the (pixel-level) specificity and \text{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 \(\alpha\) 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.

1) Pre-processing: Pre-processing is summarized in Fig. 10. Each physical map is resized from \(144 \times 172\) to \(360 \times 180\) 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
1: function PRE_PROCESS(map)
2:   ▷ Input: map to process.
3:   ▷ Output: map_{+,−} contains separate coronal hole
4:      for positively and negatively charged coronal
5:      holes of the same size as the magnetic images,
6:      with polar regions removed.
7:   ▷ Extract reference and model maps
8:   map ← extract_coronal_hole_maps (map)
9:   ▷ Remove small gaps in the maps
10:  map ← binary_close (map)
11: ▷ Separate the positive from the negatively charged
12:  coronal holes
13:  map_{+,−} ← extract_polarity_maps (map)
14: ▷ Remove regions where there are no observations,
15:   and regions with latitude 0 to 30 degrees, and
16:   150 to 180 degrees
17: map_{+,−} ← remove_regions (map_{+,−})
18: return map_{+,−}
19: 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 missing_map (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 new_map (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:

\[
\begin{align*}
\min_{m_{i,j}} \sum_{i} \sum_{j} w_{i,j} m_{i,j} & \\
\text{subject to:} & \\
\sum_{i} m_{i,j} = 1, & \quad (5) \\
\sum_{j} m_{i,j} = 1, & \quad (6) \\
m_{i,j} \in \{0, 1\} & \quad (7)
\end{align*}
\]

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
- Segmentations by two independent experts
- 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 the 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 Spec = 1), (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 want to avoid missing valid coronal holes. As a result, we prefer over-segmentation as opposed to under-segmentation. 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 [V]. 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 (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 \((\text{Sens}=1, \text{Spec}=1)\). 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, \(1\times\) refers to the performance of the proposed method. A value of \(2\times\) implies that the error was twice as large as the proposed method.

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

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

We present classification results over the training set in Table IV.

Table IV summarizes the results over the remaining 30% of the data that were left for testing. Overall, the classifier had an accuracy of 95.5%.

We also examine the relative importance of each feature in

TABLE IV: Coronal hole matching results. \textit{New} refers to coronal holes that appear in the physical model but are not seen in the consensus segmentation maps. \textit{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   |
the Random Forest classifier and report the results in Fig. 17. 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 new solar image analysis methods to support: (i) accurate segmentation of coronal holes, (ii) reproducible classification of physical models based on a manual protocol, (iii) coronal hole clus-
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.

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. We would also like to acknowledge the support of Dr. C. Henney who provided his segmentation method.
Venkatesh Jatla

Venkatesh Jatla received his B.Tech. degree in electrical and communications engineering (ECE) in 2011 from VIT, Tamilnadu, India and the M.S. degree in computer engineering in 2016 from University of New Mexico, Albuquerque, USA. He is currently working on his Ph.D. in computer engineering with Dr. Marios Pattichis. His current research interests include video analysis, video compression, image processing and machine learning. He is also interested in real time video delivery over http and webRTC.

Marios Pattichis

Marios Pattichis (M’99, SM’06) received the B.Sc. (High Hons. and Special Hons.) degree in computer sciences and the B.A. (High Hons.) degree in mathematics, both in 1991, the M.S. degree in electrical engineering in 1993, and the Ph.D. degree in computer engineering in 1998, all from the University of Texas, Austin. He is currently a Professor and Associate Chair with the Department of Electrical and Computer Engineering, at the University of New Mexico (UNM) in Albuquerque. His current research interests include digital image and video processing, video communications, dynamically reconfigurable hardware architectures, and biomedical and space image-processing applications. Dr. Pattichis is currently a senior associate editor with the IEEE Transactions on Image Processing. He has served as a senior associate editor of the IEEE Signal Processing Letters, an associate editor for the IEEE Trans-actions on Image Processing, IEEE Transactions on Industrial Informatics, and as a guest associate editor for the IEEE Transactions on Information Technology in Biomedicine. He was the general chair of the 2008 IEEE Southwest Symposium on Image Analysis and Interpretation. He was a recipient of the 2004 Electrical and Computer Engineering Distinguished Teaching Award at UNM. For his development of the digital logic design labs at UNM he was recognized by the Xilinx Corporation in 2003 and by the UNM School of Engineering’s Harrison faculty excellent award in 2006. He was a founding Co-PI of the COSMIAC research center at UNM. At UNM, he is currently the director of the image and video Processing and Communications Lab (ivPCL).

Charles Nick Arge

Charles Nick Arge received his B.S. in Physics at the University of Arizona in 1985, an M.S. degree in Physics at the University of Minnesota in 1988, and a Ph.D. in Physics in 1997 at the University of Delaware. He worked for the University of Colorado & NOAA/Space Environment Center (now Space Weather Prediction Center) from 1996-2003. From 2004-2016 he worked at the Air Force Research Laboratory (AFRL), Space Vehicles Directorate. In late 2016 he moved to NASA Goddard Space Flight Center where he currently serves as the Chief of the Solar Physics Laboratory, which is comprised of more than 100 staff and affiliates. Dr. Arge is also an Adjunct Professor in the Department of Physics and Astronomy at the University of New Mexico and in Department of Astronomy at New Mexico State University. He is a member of the American Astronomical Society (AAS), the AAS Solar Physics Division, and the American Geophysical Union. Dr. Arge received the NOAA Research’s Outstanding Scientific Paper Award in 2004. He received AFRL Awards for Technology Transfer Achievement in 2006 & 2008, Senior Leadership in 2010 & 2014, and Team Publication in 2015. He received an AFRL Outstanding Mentor Award in 2014 and a Federal Executive Board Public Service Award: Professional, Administrative, Technical in 2015. He was the lead of three Air Force Office of Scientific Research Star Teams. Dr. Arge does basic and applied research in the areas of coronal and solar wind modeling. He is co-developer of the Wang-Sheeley-Arge (WSA) coronal and solar wind model, which currently runs operationally at the NOAA National Centers for Environment Prediction (NCEP).