Multi-Modal Object Tracking and Image Fusion With Unsupervised Deep Learning

Nicholas LaHaye, *Member, IEEE*, Jordan Ott, Michael J. Garay,
Hesham Mohamed El-Askary, *Member, IEEE*, and Erik Linstead, *Senior Member, IEEE*

***Abstract*—The number of different modalities for remote sensors continues to grow, bringing with it an increase in the volume and complexity of the data being collected. Although these datasets individually provide valuable information, in aggregate they provide additional opportunities to discover meaningful patterns on a large scale. However, the ability to combine and analyze disparate datasets is challenged by the potentially vast parameter space that results from aggregation. Each dataset in itself requires instrument-specific and dataset-specific knowledge. If the intention is to use multiple, diverse datasets, one needs an understanding of how to translate and combine these parameters in an efficient and effective manner. While there are established techniques for combining datasets from specific domains or platforms, there is no generic, automated method that can address the problem in general. Here, we discuss the application of deep learning to track objects across different image-like data-modalities, given data in a similar spatio-temporal range, and automatically co-register these images. Using deep belief networks combined with unsupervised learning methods, we are able to recognize and separate different objects within image-like data in a structured manner, thus making progress toward the ultimate goal of a generic tracking and fusion pipeline requiring minimal human intervention.**

***Index Terms*—Big data applications, clustering, computer vision, deep belief networks (DBNs), deep learning.**

I. Introduction

MANY Earth-pointing, remote sensing instruments provide imagery; each with their own foci, strengths, weaknesses, and experts who use the data. These instruments acquire data in different spectral bands; have different resolutions, different orientations, and even fly on vastly different platforms. Some satellite-based datasets represent decades of mission data, while others from airborne platforms may only have a few days of campaign data. As our understanding of the geosciences grows, there is an inherent need to see how the parameters measured by one instrument can be fused with those measured by other instruments, in order to formulate a global perspective on subjects like state changes, catalysts, and system interactions. In order for scientists to properly and effectively combine different datasets, they need to have an understanding of each instrument, dataset, and the processing techniques used. When considering many instruments and different applications, this becomes burdensome and can discourage large-scale collaboration among instrument science teams. There is a need for a generic, intelligent algorithmic pipeline that can ingest datasets of different kinds and combine them in useful ways, requiring as little instrument-specific knowledge as possible. For example, such a system should be able to track objects like clouds or bodies of water across datasets, and fuse the data where appropriate [1], [2]. It is noteworthy that since scientists often deal with different kinds of data over large scales, the user should not be expected to label data in order to train the system. Instead, the system should be able to differentiate objects on its own. This means it is not necessary to know what the objects are, just that they are different. The hypothetical system would not itself make groundbreaking findings, but it should provide an environment where scientists can do so. There has been some previous work on image fusion and object detection using remote-sensing systems, however the research that has been done is either instrument-specific or, if it is multi-modal, there is some supervised learning within the system [3]. The input data are either user-labeled, or information is provided to the learning system in a manner that makes it difficult to generalize the process into efficient, multi-modal applications [3]. For example, adding geolocation information to training data implicitly adds information about map projection, and any geo-calibration information that was used, which partially limits the freedom of the learning system when working with the data. While we are aware that this information may be useful in many other learning systems with similar goals, our goal is to be able to recognize objects of similar compositions across multi-modal datasets, and in some cases, across spatiotemporal ranges. Also, labeling the data may prove to be a daunting task if the user is using many datasets, or may not even be possible at all, given the type and quantity of data considered. While instrument-specific

Manuscript received November 17, 2018; revised January 11, 2019, March 5, 2019, and April 26, 2019; accepted May 28, 2019. Date of publication June 17, 2019; date of current version September 16, 2019. *(Corresponding author: Hesham Mohamed El-Askary.)*

N. LaHaye is with the Computational and Data Sciences Department, Chapman University, Orange, CA 92866 USA, and also with the Processing Algorithms and Calibration Engineering Group, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: lahay100@mail.chapman.edu).

J. Ott is with the Computational and Data Sciences Department, Chapman University, Orange, CA 92866 USA (e-mail: ott190@mail.chapman.edu).

M. J. Garay is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: michael.j.garay@jpl.nasa.gov).

H. M. El-Askary is with the Center of Excellence in Earth Systems Modeling and Observations and the Schmid College of Science and Technology, Chapman University, Orange, CA 92866 USA, and also with the Faculty of Science, Department of Environmental Sciences, Alexandria University, Alexandria 21522, Egypt (e-mail: elaskary@chapman.edu).

E. Linstead is with the Schmid College of Science and Technology and the Machine Learning and Assistive Technologies Lab, Chapman University, Orange, CA 92866 USA (e-mail: linstead@chapman.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2019.2920234

systems do provide useful insights into the deeper information content within a single dataset, a generic system is needed to provide insights by fusing datasets. To help solve this problem, we are developing a system that relies on unsupervised learning methods to understand the structure of multi-modal data, rather than the actual content. It leaves the content specifics to an expert, or an expert system, but its intent is to provide an algorithmic pipeline where the expert can combine datasets and make multi-dataset findings with ease.

To make progress toward a generalized fusion architecture, in this paper, we apply unsupervised deep learning to the problem of object recognition and separation, given image-like data from different remote sensing platforms. Our study involves testing how well our system can separate objects within each dataset, separately. To do this, we train a deep belief network (DBN) for each instrument dataset input. This allows for a non-parametric, deep "machine understanding" of the datasets, which provides a precise dataset-wide classification technique. This not only has uses on its own, but can also be input to further learning and analysis techniques to generalize and classify over multiple datasets at a time. Unsupervised machine learning (clustering) was used here in order to translate the DBNs output to be human understandable, and to verify that these separate networks could provide enough detailed information to properly separate and recognize different objects. However, this paper is the initial step in building a skilled classifier, able to track objects across multi-modal datasets. The cluster signature our model creates key on specifics within the data unseen to the human eye and valuable to precise tracking. To test our proposed methodology, we leverage data from a heterogeneous set of imagers, namely, MISR, MODIS, Hyperion, and LandSat-8, in order to show multi-modal efficacy. The following sections will show that our models are able to separate objects within the data on a very fine scale, and even determine subsets of similar objects, like clouds.

While there has been much work in the areas of object detection and data fusion [4], our work differentiates itself from prior models in the following two significant ways: 1) its aim is generic, i.e., able to perform multi-instrument data fusion and 2) it is completely unsupervised. Because our system aims to assist with both data fusion and object detection/tracking, we have reviewed related works in both areas, splitting the object detection/tracking into supervised and unsupervised methodologies. Though we believe our method to be unique in providing multi-instrument data fusion for object recognition, many others have performed spectral clustering and image fusion. The major differences between our model and others are that other models use a single data source [5], supervised machine learning methods [6]–[9], or provided geospatial information as input while ours do not [9]–[11]. In regards to the object detection and tracking aspect of our model, while object tracking techniques using supervised learning are very successful, they require large amounts of data labeling in the offline training phase [11]–[15]. There have also been many successful unsupervised object detection/tracking projects, however, none of the proposed models for multi-modal data are completely unsupervised, instead leveraging supervised classification [16]–[19]. The previously mentioned projects have been mostly successful for what they were intended to do, however, these methodologies do not work to provide a more general model that can take into account multi-modal data, fuse the information, and then, use that to detect and track objects across these datasets. In other words, our model will be quite robust and scalable once finalized. They do not work as a solution for this project due to either a need for labels, a lack of ability to easily generalize across multiple datasets, a requirement of geospatial information as input, or a combination of all of these factors.

Section II describes the data, tools, and methods used, while Section III provides some results and discusses validation techniques. Finally, Section IV details our future work, followed by conclusions in Section V.

II. Data and Methods

Here, our goal is to recognize the separation between different objects within image-like datasets from different instruments. For the initial test set, we wanted to use a dataset that would prove efficacy in a multi-instrument domain while still providing a representative level of difficulty. With this in mind, we chose to test on the multi-angle imaging spectroradiometer's (MISR) dataset. After initial successes with MISR, we chose to also test on the Hyperion, LandSat-8, and the Moderate-Resolution Imaging Spectroradiometer (MODIS) instruments.

A. Data and Tools

The software was developed with Python 3.4.3. All of the DBN training and testing was implemented using Lrn2 Deep Learning Framework [20], utilizing Theano with a GPUArray backend [21]. The hardware utilized was an NVIDIA GeForce Titan X GPU with 12 GB memory. As for the clustering, it was performed using Scikit-Learn [22] on a machine running Ubuntu 14.04.5.

MISR, an instrument on the Terra platform, provides us with nine different cameras, one which points nadir. The other eight are split into two equal groups of forward and aft cameras. Each group has cameras that point at matching angles relative to the local normal at the Earth's surface: 25.8°, 45.6°, 60.0°, and 72.5°. In order for MISR to capture a scene with all nine cameras, it takes about 9 min [23]. This results in a continuous multi-angle image set. Doing this, we have nine images whose imagery overlaps (but does not exactly match) in space, but because each camera views the scene from a different angle, and at a slightly different time, it provides a perfect test bed for initial experiments for data fusion [1], [2].

Along with using MISR data as a test, we wanted to include multiple different instruments to prove multi-modal efficacy within this initial step. MODIS, an instrument onboard the Terra platform, was the first to be used. MODIS is an imager with 36 spectral bands whose resolutions span from 250 to 1000 m. For this test, we used only the 1000-m bands that are measured continuously, at both day and night [24]. LandSat-8 platform on which two imaging instruments sit, whose data are combined into one LandSat-8 dataset was then added to the test bed. These two instruments are the operational land-imager

TABLE I
INPUT DATASET INFORMATION

Instrument	Dataset Level	Spatial Resolution	Spectral Resolution
MISR Nadir Camera	L1A	275m	4
MISR Off-Nadir Cameras	L1A	1.1km	4
MODIS	L1A	1000m	16
Hyperion	L1Gst	30m	220
LandSat-8	L1GT	30m	11

(OLI) and the thermal infrared sensor (TIRS). The OLI has nine visible spectral bands, and TIRS has two infrared bands [25]. The last instrument we chose to test on is Hyperion, which is a high-resolution hyperspectral imager aboard the Earth-Observing-1 (EO-1) satellite. Hyperion has 220 contiguous spectral bands, all of which we use in our test [26]. Table I provides more information about the datasets used in this study.

In order to compare our results against datasets that are considered operationally viable, we chose to use MISR's support vector machine (SVM) classifier dataset. This dataset is created by inputting MISR L1B2 data into an SVM. This classifier is able to efficiently distinguish between clouds, aerosols, water, land, smoke/dust, and snow/ice with an impressive global accuracy of 81% over all defined classes [27]. While the goals of this dataset differ from those of our research, the comparison provides useful information because it allows us to verify that our system is understanding the data's structure properly in a general sense, and ensure that our data provides a different, but equally valuable service. Hence, we consider this as an initial validation step to our classes. The SVM classifier dataset aims to separate data into distinct user-defined groups using specific keys in MISR data features, including radiance information in multiple bands from multiple cameras as well as angular information, in order to provide a separation of states to the MISR data's end user. Also, as input to the training set, labels are given, as with all supervised learning systems, in order for the SVM to correctly distinguish between the desired groups. This approach should lead to a separation of labels in a manner that may prove to be more precise within the MISR dataset, given that our aim is to look at multiple instruments. Our goal differs in the sense that we want our system to gain its understanding of the data's structure, with little instrument specificity and no user-defined labels, hence, the unsupervised approach. Such system will provide a high level of robustness and scalability once implemented and will able to skillfully track objects, like plumes, across multi-modal datasets.

In doing this, we cannot tell the user what the label of an output group is, but we should be able to apply the method to remote-sensing data in general, and treat the machine-understood structure of the data as a set of separate objects, from which it can decipher matches among multi-modal data. The distinctions made from the SVM can be used in a similar sense, that is to say, one could track groupings of cloud, dust, etc., across imagery, but our goal is to be able to differentiate between cloud (or land, etc.) groupings, with some certainty, and be able to track those different groups across datasets.

As there are different levels of data processing pipelines that provide output from remote sensing systems, each with their own inherent benefits, we had to choose a single processing level to focus on. With this in mind, we chose to use the L1B1 radiance data. This dataset consists of a best estimate of the spectral radiances that represent instrument response. These radiance sets can be viewed as "as-raw-as-possible" images from each camera. There is no camera co-registration, geolocation, or geometric calibration done on this dataset. The decision was made to use the data at this level because we want to be able to be as flexible as possible in terms of including different instruments and modalities. Additionally, we wanted to capture as little instrument-based processing/algorithm information in our input data as possible. While we understand that every instrument will have a different set of caveats and dataset quirks, this allows us to mitigate biases that the system may pick up via data adjustments/calibrations, and focus simply on the data itself. For some of these instruments, there is no L1B1 dataset, only a geocorrected L1B2 dataset, and an L1A dataset, consisting of raw digital counts. For this case, we use the L1A data in order to keep with our as-raw-as-possible strategy.

To achieve its fusion goal, our methodology needs to be able to understand data structure, output its representation of the data, and then, be able to decipher different parts of that representation on its own. Because the goal is to be able to provide a framework that can easily handle many different, large datasets, and can re-train itself in an online manner, we chose to use unsupervised learning methods. It should be noted again out that our goal here is not to be able to identify a label for the groupings that are generated by the system, because that would require supervised methods. Our goal here is to locate different structures within image-like data that can be analyzed by themselves, or fed into further analysis and learning techniques. With that goal in mind, we chose a DBN as our technique to recognize the data's structure. Given images as input, the image is split up into small chunks and fed to the DBN, and the output of the network is then fed into an agglomerative clustering algorithm, which aims to separate out the different object types within the input images. Each modality has its own DBN, and the output of each DBN is fed to the clustering algorithm. Section II-B discusses in detail the current methods used and the reasoning behind them.

B. Methods Used

We began our tests by collecting the desired data from the proper NASA and USGS online data portals. For each separate instrument dataset, we read in all of the training data, accounted for fill values and values noted by the metadata to be unnatural. Then, we created 5 × 5-image chunk, to preserve some spatial information from the overarching data, which were used as the training instances. Each instance contained matching image

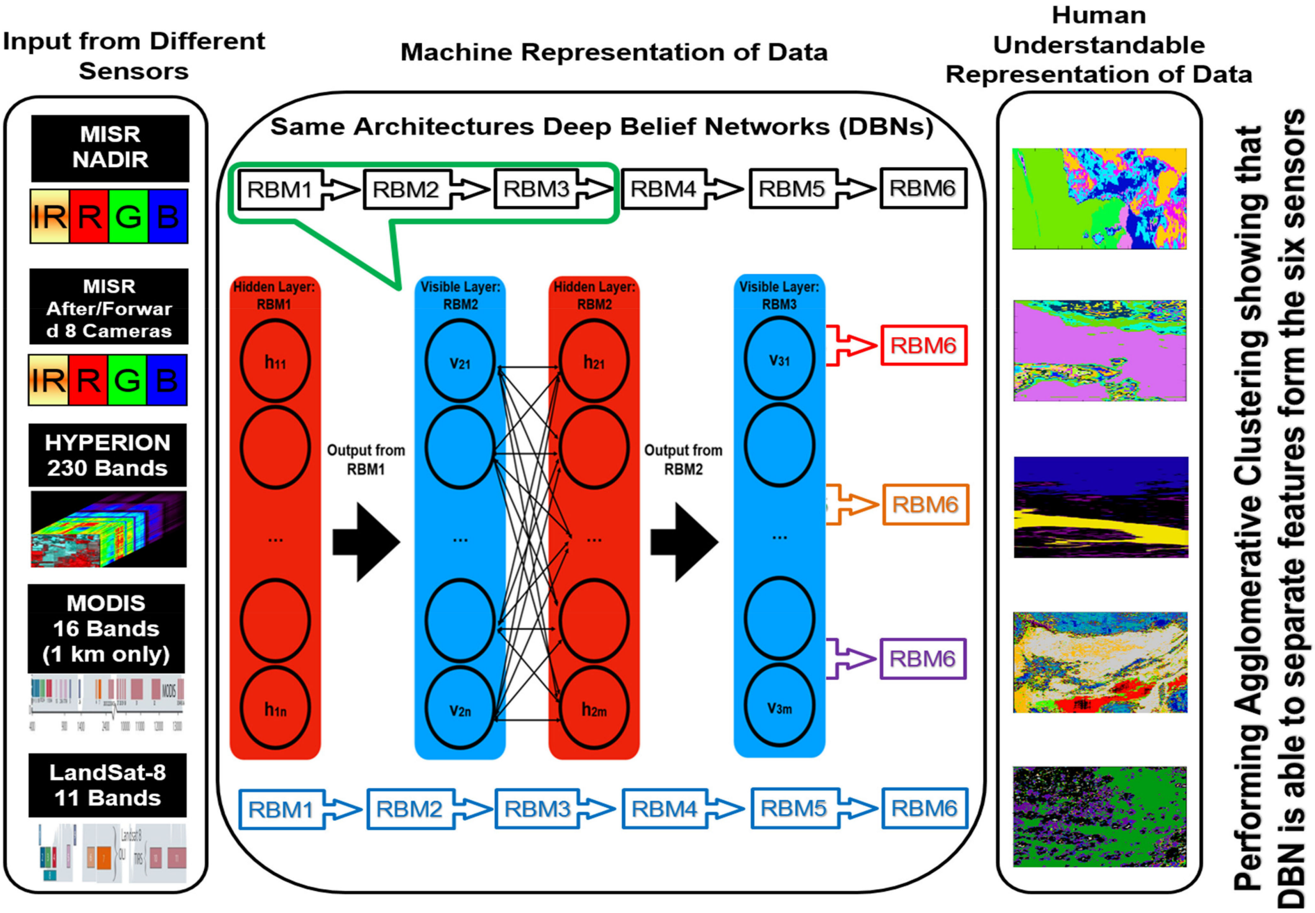
Input from Different Sensors
Machine Representation of Data
Human Understandable Representation of Data
MISR NADIR
IR R G B
MISR After/Forward 8 Cameras
IR R G B
HYPERION 230 Bands
MODIS 16 Bands (1 km only)
LandSat-8 11 Bands
Same Architectures Deep Belief Networks (DBNs)
RBM1 RBM2 RBM3 RBM4 RBM5 RBM6
Hidden Layer: RBM1
h11
...
h1n
Output from RBM1
Visible Layer: RBM2
V21
V2n
Hidden Layer: RBM2
h21
h2m
Output from RBM2
Visible Layer: RBM3
V31
V3m
RBM6
RBM6
RBM6
RBM1 RBM2 RBM3 RBM4 RBM5 RBM6
Performing Agglomerative Clustering showing that DBN is able to separate features form the six sensors

Fig. 1. Depiction of the experimental setup. Each instrument, or set of similar instruments, has their own DBN. The output of the network is then made human readable via agglomerative clustering, and the results are compared against browse images and other classification datasets. In the center of the figure, there is also a depiction of how the DBN training takes place. Output from the hidden layer of one RBM is fed into the visible layer of the next RMB, and this process is repeated until the final RBM's output is fed into the clustering algorithm.

chunks from all desired bands. We tested other sizes of chunks for data instances, but smaller chunks did not provide enough information about a data point's neighborhood within the image, and thus, provided the output of lower quality. Similarly, instances of larger chunk sizes created output that were of too low a resolution to be of value. The data's dimensions were then translated to match what the DBN required as input, and it was input to the DBN as training instances. After the network was trained, the same data setup methods were employed for the test sets of data.

A DBN is a composition of restricted Boltzmann machines (RBM) that can be trained in a supervised or unsupervised fashion [26]. An RBM is a variation of a hidden Markov random field whose energy function is linear in its free parameters, and whose edges are restricted to only being able to make connections between adjacent layers. In this way, visible units cannot connect to visible units and non-visible units cannot connect to non-visible units [28].

The energy function of an RBM is

$$E\left(v,\, h\right) = \,-b^T\, v\, -\, c^T\, h\, -\, hW\, v \tag{1}$$

where v is the set of visible units, h is the set of hidden units, b is the set of offsets of the visible units, c is the set of offsets of the hidden units, and W represents the weights between the hidden and visible units. This can be translated into the free energy formula as follows:

$$F\left(v\right) = -b^T v - \sum_i \log\left(\sum_{h_i} e^{h_i(c_i + W_i v)}\right) \tag{2}$$

which allows us, by definition of energy-based models with hidden units, to define the probability distribution as

$$P = \frac{e^{-F(x)}}{\sum_x e^{-F(x)}}. \tag{3}$$

In order to train a DBN, each RBM is trained separately, where the first RBM takes in the input, and the others use the previous RBM's output as input. Instead of using gradient descent on the second derivatives of the negative log likelihood equation, a process called contrastive divergence is the preferred method to train RBMs. This is because in order to use the typical gradient descent method, you must run each RBM (a Markov field) until it converges on its equilibrium distribution in order to use expected values from that distribution to calculate the parameter updates. This is both computationally expensive and the variance on the values sampled from the equilibrium distribution is usually high, causing issues when training. Contrastive divergence, on the other hand, instead of comparing the initial (input) and equilibrium distributions, it runs an initial *N* number of Gibbs sampling steps, and then, uses KL-divergence as a measure to keep the distributions created by future update steps from straying too far from the initial distribution, [29]. Along with this, [30] proposes using bias constraints on the contrastive divergence method in order to account for the sparsity and selectivity of a given set of hidden units. In order to achieve activation diversity among a set of hidden units, a single hidden unit should only activate when necessary, not just every time an instance reaches its level in the network, hence, sparsity. With the same end in mind, a set of hidden units, while sparsely activating, should not always activate at the same time, hence, selectivity. RBMs singularly can recognize unique distribution patterns within images and, when multiple RBMs are connected together, can provide a useful technique for tracking complex features within a dataset. This provides the perfect framework for learning object and image similarity in an unsupervised fashion.

For this experiment, we chose to use a simple architecture of six RBMs for each dataset, as depicted in Fig. 1. Initially, we used a network consisting of three, but found that it did not provide enough feature recognition when studying the final output. We then increased the number of RBMs until we found the smallest set that resulted in an output that we believe is detailed enough. The network is trained in a layer-wise fashion, with the output of the prior RBM being used as the training input of the next RBM.

Because an end-user cannot necessarily interpret the output of one or more RBMs, we need a way to restructure the data into human-recognizable form; this is why the agglomerative clustering was chosen as a machine understandable to a human understandable translation technique. A general clustering problem can be seen as a multi-objective optimization problem. The input is a set of *N* data points with *M* features. The goal is to group the data into a desired number of clusters, *K*, while minimizing the given error, or distortion, function. In agglomerative, or hierarchical, clustering, each data point beings in its own cluster $\mathrm{s_j}$. At each step, all clusters are compared, and a merge operation is performed $\mathrm{s_a} = \mathrm{s_a} \cup \mathrm{s_b}$, where *a* and *b* are just cluster indices at step *i*. This merge operation is performed on the two clusters whose merge minimally effects the error function. There are many error functions to use, but we found Ward's method to work best for us. Ward's method aims to minimize variance among a cluster, so it punishes merges that increase variance among a cluster with the function:

$$\Delta\left(A, B\right) = \Sigma_{i\epsilon A\cup B}\ \left|\left|X_i - M_{A\cup B}\right|\right|^2 - \Sigma_{i\epsilon A}\left|\left|X - M_A\right|\right|^2 - \Sigma_{i\epsilon B}\left|\left|X - M_B\right|\right|^2 \quad (4)$$

$$= \frac{N_A N_B}{N_A + N_B}\left|\left|M_A - M_B\right|\right|^2 \quad (5)$$

where *A* and *B* are two clusters, $\mathrm{X}\varepsilon\mathrm{A} \cup \mathrm{B}$, $\mathrm{M_A}$ is the center of cluster *A*, $\mathrm{N_A}$ is the number of data points in cluster *A*. For large datasets, this method can be difficult unless a connectivity matrix is used. The connectivity matrix defines how instances are connected within the actual dataset, so the algorithm does not necessarily have to calculate the distortion penalty for each combination of clusters. In our case, we used a *K*-nearest neighbor graph. This method only calculates the merge cost between a cluster and the clusters of its *K* nearest neighbors [31]. We also tested error functions that aim to minimize the average, minimum, and maximum distances between a cluster and the clusters of its *K* nearest neighbors, but found that Ward's method gave us the best results.

III. Results

We began our tests by running the clustering algorithm on our raw data in order to create a baseline on which we could measure success. While the clustering itself does well in capturing the general structure of the data, it does not capture enough detail within the data to successfully move on to further states of fusion. Next, we input the images into the DBNs we have trained, one for the AN camera images, at 257-m resolution for all spectral bands, and one for the other eight camera's images, at 1.1-km resolution for all spectral bands. While the raw data for all of the cameras, except AN have the red band data at 257-m resolution, we downscaled the red band image to match the other spectral bands within the respective cameras' images for uniformity of the input data. For training data, we used roughly 1 200 000 image chunks per DBN with all four channels used in each instance. For testing, we used eight sets of roughly 30 000 image chunks, again with all four channels. The training data and testing data were taken from different files, or scenes, to ensure mutual exclusivity. The training set was chosen to be large enough to be representative of different kinds of scenes, but small enough to be validated in the manual/visual way we describe later, given the nature of this task at its current stage. Once we obtained the output of the test data from the DBNs, we put all output of each DBN into the clustering algorithm. We kept the data from each DBN separate at this point because we wanted to see if the instrument-specific DBNs were learning the data format correctly. The plan in the future is to combine this data in an instrument-generic DBN at the tail end of all instrument-specific networks.

In order to validate the output of our test, we compared the data to the MISR SVM classifier dataset, the input image, and a RGB MISR image. We also do agglomerative clustering on the input imagery itself, to make sure that the DBN is itself learning the structure of the data, and the value is not just from the clustering. Unfortunately, there is no classifier product at the L1B1 level of MISR data. The L1B2 MISR data are re-projected from the image grid of lines and samples on to the SubOrbital Mercator (SOM) grid. Also, the classifier data are 1.1-km resolution for all images. This makes comparison to the MISR L1B1 data a bit inexact, but because this task can be handled visually, we can measure success easily. For the task, we are trying to accomplish, learning, in detail, the structure of the data, including different objects, and subgroups within objects, we can see that our techniques perform very well.

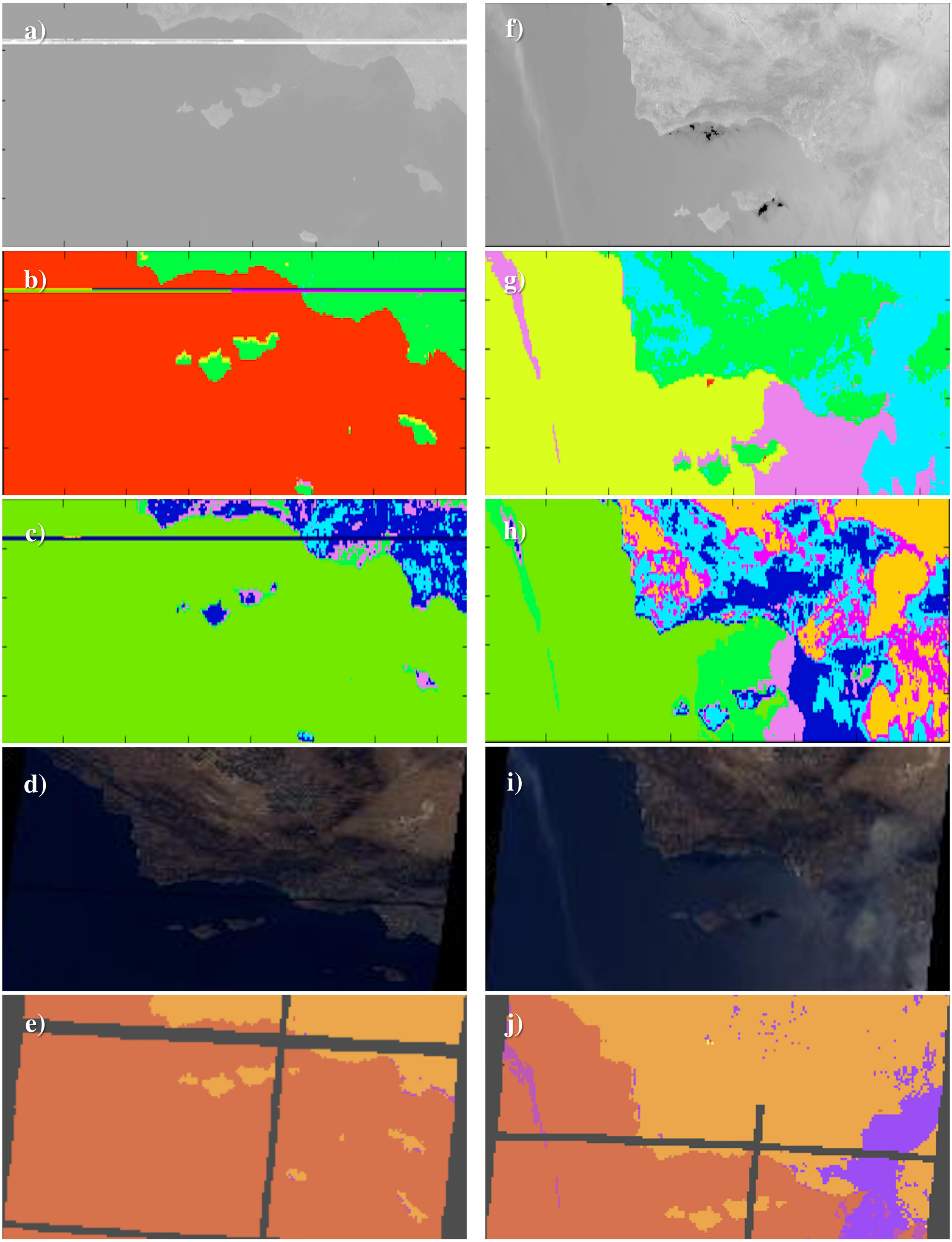

Fig. 2. Two column-wise cases of inputs and results from the MISR AN camera's DBN. (a) and (f) depict one band of input. (b) and (g) are the result of performing agglomerative clustering on the input imagery alone. (c) and (h) show the results of performing agglomerative clustering on the output of the DBN. (d) and (i) are reference RGB imagery. (e) and (j) are the MISR SVM classifier output for the same image area.

It should be noted that the colors utilized to depict clustering results do not represent the same clusters between tests or instruments. That is to say, for example, while Fig. 2(c) and 2(h) share colors, Fig. 2(b) and 2(g) do not, all four result sets mentioned have been clustered separately, and thus, the colors do not have cross-result meaning. They are only representative of separate clusters within a single result image.

We can see in Figs. 2 and 3 that the combination of the DBN and clustering capture the structure of the data very well. Among the different test sets, we are able to correctly differentiate land, water, and clouds. Furthermore, we can see, especially in the cases where the spectral resolution is high, that the agglomerative clustering of the input data alone does not produce readable, well-structured data, like the output that comes from the DBN and clustering used together. Along with this, it looks like we have maintained similar general structure as that of the SVM classifier dataset, and along with that, the system has learned a detailed structural knowledge of the imagery, one it can map across multiple images, most of which can be matched to what the eye can see. The structure includes many mixtures of subtypes within the features that are visible to the naked eye. This is expected, as there are often different mixtures of, land, cloud, aerosol, and other factors together in a single location. This detailed understanding of the data does not necessarily have to be exactly the same one that a human would have, as the machine is learning on its own what are “important” features, but if it can continue to differentiate between object sets, and use that to track those objects across imagery as it has in this test, then this method will continue to prove successful.

In many cases, visual validation of clustering results proves to be sufficient [32], yet, here, we combined our qualitative analysis of cluster results with a quantitative one. Since, as is typical in unsupervised machine learning studies, we do not have ground truth data to compare against, the evaluation can depend only on the model itself to determine effectiveness. Therefore, we chose to quantitatively evaluate the clustering performance before and after the data were passed through the DBN. To verify the correctness, we used the average silhouette width [33] to measure how well defined the generated clusters are. To calculate the silhouette width, $\mathrm{s(i)}$, for each observation, *i*, we first need to calculate $\mathrm{a(i)}$, the average distance between *i*, and all other points in its cluster, $\mathrm{b(i)}$, the average distance between *i* and all points in the nearest cluster besides its own. The silhouette width is then calculated as

$$s\left(i\right) = \frac{a\left(i\right) * b\left(i\right)}{\max\left\{a\left(i\right), b\left(i\right)\right\}}. \tag{6}$$

The silhouette width allows us to measure the compactness and separation of clusters, where more compact and disparate clusters, or a generally better performing model, relate to a higher silhouette width, in the valid range of $[-1, 1]$. This value can be averaged over a single cluster, or over whole scenes to measure, whether the clusters in question are well defined or not.

For each of the DBNs used in this study, we averaged the silhouette score of the clustering results of the test data before it is input to the DBN, and then, again on the clustering results from the output of the DBN. In general, an increase in score represents a partitioning of the data in a way that more closely represents latent structure encoded in it. From the results shown in Table II, we can see that there is a significant improvement with respect to all instruments after the data passes through the DBN.

TABLE II
SILHOUETTE WIDTH AVERAGE VALUES

Instrument	Pre-DBN	Post-DBN
MISR Nadir Camera	0.2024	0.5496
MISR Off-Nadir Cameras	0.2517	0.5833
MODIS	0.13477	0.73204
Hyperion	0.16620	0.5102
LandSat-8	0.2361	0.6228

For example, for both the MISR and Landsat-8 instruments, the score climbs by more than a factor of two. For MODIS, the score climbs from 0.13477 to 0.73204, a factor of over five. Even MISR Nadir, which exhibits the smallest increase in score, achieves a 2.32x increase. Thus, even in the absence of the labeled truth data, we are able to demonstrate the efficacy of our DBN pipeline to effectively cluster data, regardless of the instrument. The final silhouette values of each instrument/instrument set are correlated with the spatial and spectral resolution of the datasets. The Hyperion DBN output performs the worst, and this is due to the extremely high spectral and spatial resolution. The specificity of the data allows for similarities to be observed between entities that would be clearly distinct in lower resolution data. In terms of performance, the next two DBN output sets are the MISR DBN output sets. These output sets perform at the lower end of the performance scale within this experiment due to the exact opposite reason: The information within their dataset is not specific enough to create more distinct clusters. The MISR off-nadir DBN output performs slightly better than the output of the MISR nadir DBN output due to the lower spectral resolution. Next, the LandSat-8 DBN output performs slightly worse than the MODIS DBN output. This is due to the fact that the LandSat-8 data, while having a slightly lower spectral resolution than the MODIS data, has a significantly higher spatial resolution. Hence, the Landsat-8 and MODIS datasets appear to be within the range of most-desired spatial and spectral resolution ratios for optimal performance in this context.

After the initial test with the MISR data, we ran similar tests on MODIS, Hyperion, and LandSat-8 data. We used similar amounts of training and testing data as the MISR test for each of the next three tests. In Fig. 3, we have included one band of the input image, the results of clustering the input data, and the results of clustering the output of the DBN. Unfortunately, these projects do not have classification datasets like MISR does, but

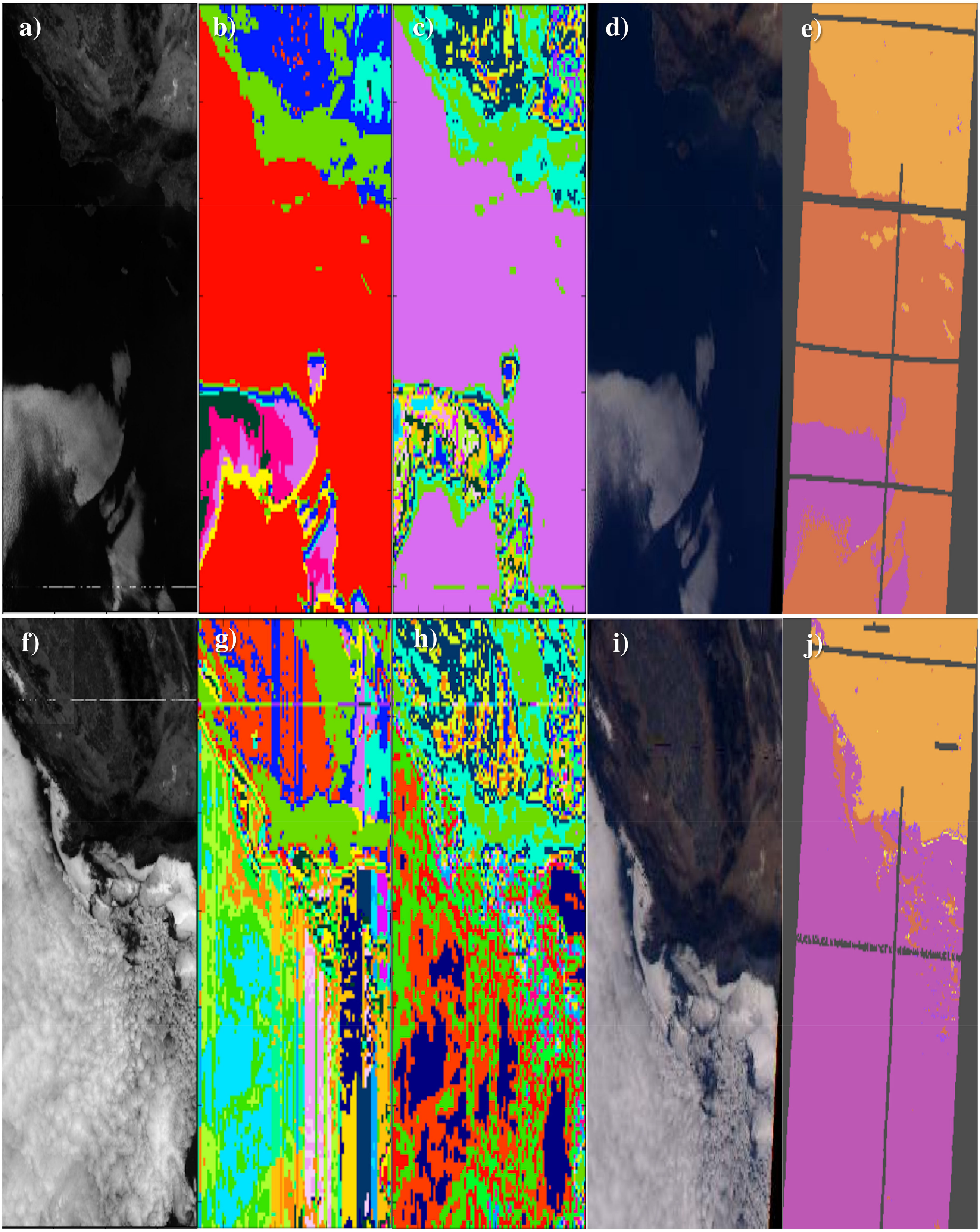

Fig. 3. Two row-wise cases of inputs and results from the MISR Aft and Forward cameras' DBN. (a) and (f) depict one band of input. (b) and (g) are the result of performing agglomerative clustering on the input imagery alone. (c) and (h) show the results of performing agglomerative clustering on the output of the DBN. (d) and (i) are reference RGB imagery. (e) and (j) are the MISR SVM classifier output for the same image area.

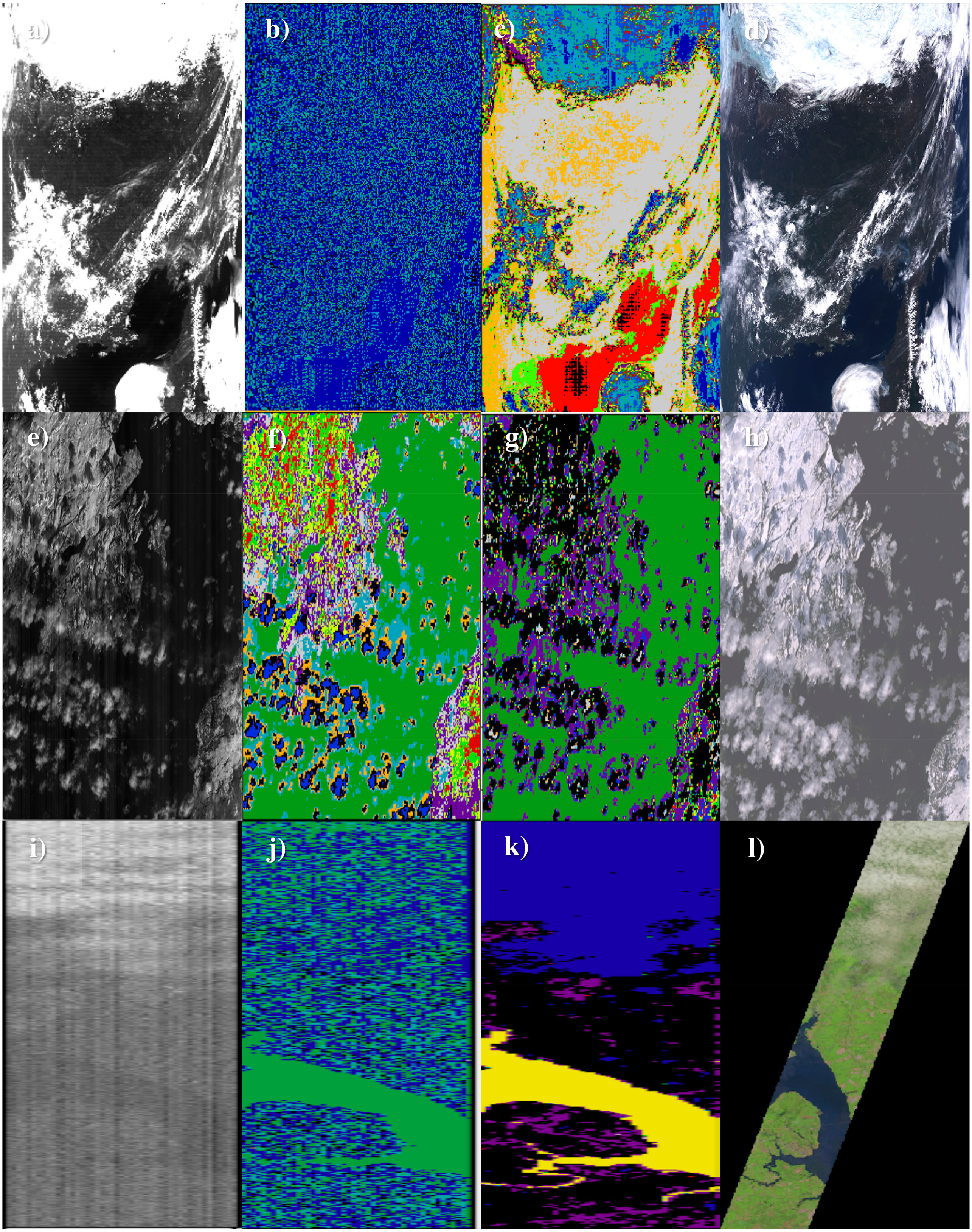

Fig. 4. One row-wise case of inputs and results for the MODIS, LandSat-8, and Hyperion DBN, respectively. (a), (e), and (i) depict one band of input. (b), (f), and (j) are the result of performing agglomerative clustering on the input imagery alone. (c), (g), and (k) show the results of performing agglomerative clustering on the output of the DBN. (d), (h), and (l) are reference RGB imagery.

we can still use visual tests, alongside the silhouette width averages, to surmise the efficacy in these cases. For that, we looked at other examples as shown in Fig. 4. In the case of the MODIS example, there is a clear distinction between land, water, and clouds. The coastline is identified accurately, and the different cloud formations are also correctly recognized, even when their shapes are seemingly hard to capture. In the case of the MODIS data being clustered pre-DBN, we can see that there is really only a distinction between water and everything else, and even then, there are groups of pixels classified in incorrect spots, and the rest of the image is completely indecipherable. The Hyperion case is very similar, with the output of our system finding the cloud system in a realistic form, locating the entirety of the body of water, and also separating different kinds of land. Again, the only thing that is decipherable in the pre-DBN clustering case for Hyperion is the body of water, which does not even appear to be outlined as accurately as after the DBN is used. In the last case, using Landsat 8 data, the biggest distinction that the DBN helps with, is finding the clouds over both land and water, which the input clustering is not quite able to do, which is an important factor when trying to track objects like cloud systems. It also appears to find the coastline more accurately.

IV. Future Work

Our next steps include taking the output from the multiple instrument-specific DBNs and combine them as input to a generic DBN. From here, we plan to run similar tests. This will allow us to not only be multi- modal, but to have an output that is equivalent among all input sets. This is an important step because it will help to remove some of the different instrument-specific intricacies each separate RBM may have learned that detract from the final cohesive output. For a validation step, we can also test our system on datasets that intersect with the flight line of the CALIOP satellite, in order to use data from the CALIPSO instrument. The CALIPSO instrument can be used to get cloud systems' profiles as well as a classification of the cloud or aerosol type being observed. This would allow us to do further quantitative validation of both the clustering and object tracking with some of the more complex objects we are looking to work with. Finally, we will research the use of the output of all of the previous steps to track objects that are found via our system and co-locate/fuse the data where appropriate.

V. Conclusion

This research has shown that unsupervised deep learning, paired with unsupervised clustering techniques, can provide precise object-recognition and classification information. In our experiments, we were able to demonstrate that the techniques can recognize similar object subtypes across multiple images from the same instrument in an extremely detailed manner. Also, given that we trained a single network using multiple different MISR cameras, we showed that this method is at least successful for tracking objects across instrument datasets that when the instruments are very similar, and given the detailed structure in which each object is mapped in an image, tracking objects over datasets with larger differences seems possible.

Acknowledgment

The authors would like to thank the Schmid College of Science and Technology, Chapman University, for supporting this research. The authors would also like to thank the Machine Learning and Assistive Technology (MLAT) Lab, Chapman University, for the use of their computing resources and laboratory, which has also benefitted from generous hardware donations from NVidia. Without NASA and USGS providing public data archives, this research would not have been possible. Finally, the authors would like to thank the anonymous reviewers for taking the time to read this paper and provide valuable feedback.

References

[1] A. Agarwal, H. El-Askary, T. El-Ghazawi, M. Kafatos, and J. Le-Moigne, "Efficient PCA fusion techniques for MISR multi-angle observations with applications to monitoring dust storms," *IEEE Geosci. Remote Sens. Lett.*, vol. 4 no. 4, pp. 678–682, Oct. 2007.

[2] H. El-Askary, A. Agarwal, T. El-Ghazawi, M. Kafatos, and J. Le-Moigne, "Enhancing dust storm detection using PCA based data fusion," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 25–29, 2005, pp. 1424–1427.

[3] J. J. Gapper, H. El-Askary, E. Linstead, and T. Piechota, "Evaluation of spatial generalization characteristics of a robust classifier as applied to coral reef habitats in remote islands of the Pacific Ocean," *Remote Sens.*, vol. 10, no. 11, 2018, Art. no. 1774.

[4] C. Pohl and J. L. V. Genderen, "Review article multisensor image fusion in remote sensing: Concepts, methods and applications," *Int. J. Remote Sens.*, vol. 19, no. 5, pp. 823–854, 1998.

[5] L. Yang, S. Tian, L. Yu, F. Ye, J. Qian, and Y. Qian, "Deep learning for extracting water body from Landsat imagery," *Int. J. Innovative Comput. Inf. Control*, vol. 11, no. 6, pp. 1913–1929, Dec. 2015.

[6] S. Li, J. T. Kwok, and Y. Wang, "Multifocus image fusion using artificial neural networks," *Pattern Recognit. Lett.*, vol. 23, no. 8, pp. 985–997, 2002.

[7] J. Dong, D. Zhuang, Y. Huang, and J. Fu, "Advances in multi-sensor data fusion: Algorithms and applications," *Sensors*, vol. 9, no. 10, pp. 7771–7784, 2009.

[8] L. Gonzalez and X. Briottet, "North Africa and Saudi Arabia day/night sandstorm survey (NASCube)," *Remote Sens.*, vol. 9, no. 9, 2017, Art. no. 896. [Online]. Available: http://www.mdpi.com/2072-4292/9/9/896

[9] J. Zhong, B. Yang, G. Huang, F. Zhong, and Z. Chen, "Remote sensing image fusion with convolutional neural network," *Sens. Imaging*, vol. 17, no. 1, pp. 1–16, Nov. 2016.

[10] M. Mostafa, A. Farag, and E. Essock, "Multimodality image registration and fusion using neural network," in *Proc. 3rd Int. Conf. Inf. Fusion*, 2000, pp. WED3/3–WED3/9.

[11] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[12] D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker, "Machine learning in geosciences and remote sensing," *Geosci. Frontiers*, vol. 7, no. 1, pp. 3–10, 2016.

[13] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 749–765.

[14] Z. Shao and J. Cai, "Remote sensing image fusion with deep convolutional neural network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 5, pp. 1656–1669, May 2018.

[15] H. Song, Q. Liu, G. Wang, R. Hang, and B. Huang, "Spatiotemporal satellite image fusion using deep convolutional neural networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 821–829, Mar. 2018.

[16] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.

[17] X. Lu, X. Zheng, and Y. Yuan, "Remote sensing scene classification by unsupervised representation learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5148–5157, Sep. 2017.

[18] D. Lin, K. Fu, Y. Wang, G. Xu, and X. Sun, "MARTA GANs: Unsupervised representation learning for remote sensing image classification," in *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2092–2096, Nov. 2017.

[19] A. Romero, C. Gatta, and G. Camps-Valls, “Unsupervised deep feature extraction for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[20] S. Lattner *et al.*, “Lrn2,” 2016. [Online]. Available: https://github.com/OFAI/lrn2

[21] J. Bergstra *et al.*, “Theano: A CPU and GPU math compiler in python,” in *Proc. 9th Python Sci. Comput. Conf. (SciPy)*, 2010, pp. 1–7.

[22] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[23] D. Diner, “MISR Experiment Overview,” 1999. [Online]. Available: https://eospso.gsfc.nasa.gov/sites/default/files/atbd/EXPER-MISR.pdf

[24] M. D. King, Y. J. Kaufman, W. P. Menzel, and D. Tanre, “Remote sensing of cloud, aerosol, and water vapor properties from the moderate resolution imaging spectrometer (MODIS),” *IEEE Trans. Geosci. Remote Sens.*, vol. 30, no. 1, pp. 2–27, Jan. 1992.

[25] D. Roy *et al.*, “Landsat-8: Science and product vision for terrestrial global change research,” *Remote Sens. Environ.*, vol. 145, pp. 154–172, Apr. 2014.

[26] M. Folkman, J. Pearlman, L. Liao, and P. Jarecke, “Eo-1/Hyperion hyperspectral imager design, development, characterization, and calibration,” *Proc. SPIE*, vol. 4151, 2001, doi: 10.1117/12.417022.

[27] D. Mazzoni, M. J. Garay, R. Davies, and D. Nelson, “An operational MISR pixel classifier using support vector machines,” *Remote Sens. Environ.*, vol. 107, pp. 149–158, 2007.

[28] G. E. Hinton and R. Salakhutdinov, “Deep Boltzmann machines,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, vol. 5, pp. 448–455.

[29] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002. [Online]. Available: https://dx.doi.org/10.1162/089976602760128018

[30] H. Goh, N. Thome, and M. Cord, “Biasing restricted Boltzmann machines to manipulate latent selectivity and sparsity,” in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2010, pp. 419–426.

[31] P. Franti, O. Virmajoki, and V. Hautamaki, “Fast PNN-based clustering using K-nearest neighbor graph,” in *Proc. IEEE Int. Conf. Data Mining*, 2003, pp. 525–528.

[32] P. R. Kersten, J.-S. Lee, and T. L. Ainsworth, “Unsupervised classification of polarimetric synthetic aperture radar images using fuzzy clustering and EM clustering,” *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 519–527, Mar. 2005, doi: 10.1109/TGRS.2004.842108.

[33] L. Kauffman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York, NY, USA: Wiley, 1990.

Nicholas LaHaye received the B.S. degree in computer science and mathematics from Chapman University, Orange, CA, USA, in 2013, the M.S. degree in computer science, with emphasis on data science, from the University of Southern California, Los Angeles, CA, USA, in 2016. He is currently working toward the Ph.D. degree in computational and data sciences with Chapman University.

Since an internship in 2012, he has been a Software Engineer with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, working on science data processing and instrument calibration for both satellite-based and airborne instruments. His research interests include unsupervised machine learning, computer vision, and deep learning architectures.

Jordan Ott received the B.S. degree in computer science and the M.S. degree in computational and data sciences from Chapman University, Orange, CA, USA, in 2017 and 2018, respectively. He is currently working toward the Ph.D. degree in computer science from the University of California, Irvine, CA, USA.

His research interests include computer vision, deep learning, and computational neuroscience.

Michael J. Garay received the B.S. degree in physics and the B.A. degree in English Literature from the University of Toledo, OH, USA, both in 1995 and the M.S. degree in atmospheric sciences from the University of California, Los Angeles, CA, USA, in 2004.

He is currently a member of the Multi-angle Imaging SpectroRadiometer (MISR) Science Team, Jet Propulsion Laboratory (JPL), California Institute of Technology, Pasadena, CA, working on aerosol and cloud retrievals. His work focuses on algorithm improvement, testing, validation, and the dissemination of scientific results from the MISR instrument. He is also working on tasks related to the multi-angle spectropolarimetric imager, a successor to MISR being developed at the JPL. These tasks involve polarized radiative transfer simulations, data visualization, and analysis. He has also worked on a number of projects with the JPL including automatic image classification using support vector machines, multi-instrument and multi-platform data fusion, sensor webs, and feature identification and tracking. He has extensive experience in data visualization and analysis using a variety of satellite, airborne, and ground-based instruments. His research interests include one-dimensional and three-dimensional polarized radiative transfer, satellite remote sensing of clouds and aerosols and related physical processes, multi-instrument data fusion and multi-platform sensor webs, machine learning techniques, and data analysis and visualization.

Hesham Mohamed El-Askary received the B.S. degree in physics with focus on geophysics from Alexandria University, Alexandria, Egypt, in 1994, and two M.S. degrees in earth systems science and computational science and informatics and the Ph.D. degree with focus on environmental physics from George Mason University, Fairfax, VA, USA, in 2003 and 2004, respectively.

Since 2008, he has been with the Schmid College of Science and Technology, Chapman University, Orange, CA, USA, and is currently a Professor of Earth Systems Science and Remote Sensing and serves as the Program Director for the Computational and Data Sciences graduate program. He is also a Full Professor (on leave) with the Faculty of Science, Alexandria University. He has authored or coauthored more than 100 publications including peer reviewed papers, book chapters, full conference proceedings, and other conference papers. He is a Peer Reviewer for several international journals such as *Atmospheric Chemistry and Physics*, *Remote Sensing*, IEEE TRANSACTION ON GEOSCIENCES AND REMOTE SENSING, and others. His work has been funded by NASA, USDA, NSF, and EU. His research interests include modeling and observations of earth systems' processes with focus on natural disasters.

Erik Linstead received the B.S. degree in computer science from Chapman University, Orange, CA, USA, in 2001, the M.S. degree in computer science from Stanford University, Stanford, CA, USA, and the Ph.D. degree in computer science from the University of California, Irvine, CA, USA, in 2003 and 2009, respectively.

He is an Assistant Professor of Computer Science and Software Engineering with the Schmid College of Science and Technology, Chapman University, where he also serves as the Director of undergraduate computing programs, and is the Principal Investigator with the Machine Learning and Assistive Technology Lab. His research interests include the areas of artificial intelligence, machine learning, and information retrieval.

Dr. Linstead is a Senior Member of the ACM.