DeepFuseNet of Omnidirectional Far-Infrared and Visual Stream for Vegetation Detection

David L. Stone, *Member, IEEE*, Sumved Ravi, Emrah Benli, *Member, IEEE*, and Yuichi Motai, *Senior Member, IEEE*

***Abstract*— This article investigates the application of deep learning (DL) to the fusion of omnidirectional (O-D) infrared (IR) sensors and O-D visual sensors to improve the intelligent perception of autonomous robotic systems. Recent techniques primarily focus on O-D and conventional visual sensors for applications in localization, mapping, and tracking. The robotic vision systems have not sufficiently utilized the combination of O-D IR and O-D visual sensors, coupled with DL, for the extraction of vegetation material. We will be showing the contradiction between current approaches and our deep vegetation learning sensor fusion. This article introduces two architectures: 1) the application of two autoencoders feeding into a four-layer convolutional neural network (CNN) and 2) two deep CNN feature extractors feeding a deep CNN fusion network (DeepFuseNet) for the fusion of O-D IR and O-D visual sensors to better address the number of false detects inherent in indices-based spectral decomposition. We compare our DL results to our previous work with normalized difference vegetation index (NDVI) and IR region-based spectral fusion, and to traditional machine learning approaches. This work proves that the fusion of the O-D IR and O-D visual streams utilizing our DeepFuseNet DL approach outperforms both the previous NVDI fused with far-IR region segmentation and traditional machine learning approaches. Experimental results of our method validate a 92% reduction in false detects compared to traditional indices-based detection. This article contributes a novel method for the fusion of O-D visual and O-D IR sensors using two CNN feature extractors feeding into a deep CNN (DeepFuseNet).**

***Index Terms*— Convolutional neural network (CNN), deep learning (DL), object recognition, omnidirectional (O-D) far-infrared (FIR) and visual fusion, semantic extraction, vegetation detection.**

I. Introduction

THE practical use of unmanned ground vehicles (UGV) operating with small teams of humans in many civilian, service, and military applications requires a more robust method of intelligent perception for determining the possibility of the surrounding environment. The current state of obstacle detection and avoidance enables the handling of well-structured obstacles, but it is limited in its ability to distinguish between solid obstacles and low-density passable objects (grass). Through enhanced utilization of deep learning (DL) and convolutional neural networks (CNN) coupled with visual and infrared (IR) omnidirectional (O-D) camera systems, the corrections required and errors introduced are reduced when creating a 360° fusion of our visual and IR O-D cameras. In addition, we have reduced the computation requirements, making our approach more feasible for small, resource-constrained robots. We have reduced error sources by the fusion of O-D IR and O-D electro-optical cameras. The fusion of multispectral sources with reduced error minimizes error propagation forward in the perception world model. These advantages make the O-D sensors optimal tools for cost-efficient perception of robotic systems.

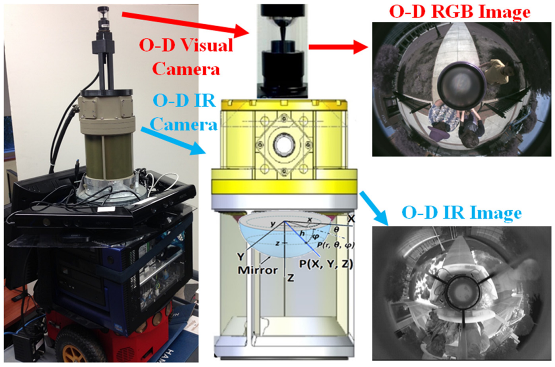


Fig. 1. Robot with O-D IR and O-D visual cameras, and onboard computer. A blow-up view of camera with geometry overlay. O-D RGB and O-D IR image from the cameras.

Fig. 1 is an example of the Pioneer robot with an O-D IR, and O-D visual camera mounted on top, with a blow-up sketch of the camera with internal mirror geometry overlaid, and the O-D RGB and O-D IR images alongside. The reason for choosing O-D vision and IR is due to their superior performance as laid out in [47].

We improve the benefits of possible low cost, O-D IR, and O-D visual sensors utilizing intelligent, DL perception algorithms to enhance the robot's vision. This article proposes a method for enhanced vegetation detection and the reduction of false detects through the DL fusion of the O-D thermal and color vision. We chose a two-stream approach rather than the traditional, stacked-band approach so that we could better learn the disparate but related features of the

Manuscript received September 4, 2020; revised November 6, 2020; accepted December 1, 2020. Date of publication January 5, 2021; date of current version October 26, 2021. This work was supported in part by the U.S. Navy, in part by the Naval Surface Warfare Center Dahlgren, and in part by the Office of Naval Research. *(Corresponding author: Yuichi Motai.)*

David L. Stone is with Virginia Commonwealth University, Richmond, VA 23284 USA, also with the Naval Surface Warfare Center, Dahlgren, VA 22448 USA, and also with the Marine Corps Warfighting Lab, Quantico, VA 22448 USA.

Sumved Ravi is with the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: sumved97@vt.edu).

Emrah Benli is with the Department of Electrical and Electronics Engineering, Karadeniz Technical University, Trabzon 61080, Turkey (e-mail: benlie@vcu.edu).

Yuichi Motai is with the Department of Electrical and Computer Engineering, Virginia Commonwealth University, Richmond, VA 23284-3068 USA (e-mail: ymotai@vcu.edu).

Digital Object Identifier 10.1109/TGRS.2020.3044487

two streams before fusing the weights in the final model. Our primary experimental results are obtained using our deep CNN fusion network (DeepFuseNet) approach from the combination of O-D thermal and color vision sensors. They are compared to our previous work with index/region-based fusion of O-D IR/visual Kinect cameras and with the O-D IR/O-D visual camera. We present the results in Section IV. We also compare to more traditional machine learning approaches.

The contribution of this work is to utilize an O-D visual sensor on top of an O-D far IR (FIR) camera by introducing a novel fusion method, named deep fusion network (DeepFuseNet). This provides for semantic scene extraction or semantic region classification and multimodal fusion of thermal and color vision of O-D sensors. Our approach uses deep transfer learning on the ImageNet large data set expanded to our smaller data set. We fuse the visual semantics with thermal region semantic structure and DL to improve the number of false detects compared to the traditional normalized difference vegetation index (NDVI) approach, thermal region fusion (TRF), and traditional machine learning approaches. One drawback of DL is the need to train on large data sets. Since we have a relatively small data set, we first train on the large ImageNet data set with over a million images and 1000 categories. We then apply transfer learning to adapt the learned weights to our smaller data set.

We organize the remainder of this article as follows. Section II discusses related works in O-D camera setting, vegetation detection, and DL-based sensor fusion; in Section III, we proposed our method on the utilization of deep CNN semantic feature extraction from the two streams using DL to extract key elements of the signatures from both the O-D vision and O-D IR cameras, followed by the application of a high-level deep fusion using DeepFuseNet. We applied two approaches to the individual camera-stream feature extraction process: first, using two autoencoder feature extractors to output the vegetation regions, and second, two deep CNN feature extractors, each feeding the final DeepFuseNet. Section IV presents the experimental setting and results of our DeepFuseNet approach in identifying the semantics of the environment's context-dependent features. Finally, in Section V, the conclusion is presented.

II. Prior Related Works

Cost-efficient operations with enhanced robotic perception are crucial for unmanned systems in applications such as rescue, military, and police. Industrial and commercial applications will also take advantage from this article. The literature review of this work focuses on the recent research with O-D sensors and the developments in mapping, localization and tracking, robot navigation, and obstacle detection, but not in the area of vegetation detection.

The fusion of IR and visual O-D sensors is valuable in order to obtain the improved robotic perception. The organization of the related works section is given as Section II-A O-D camera setting, Section II-B visual and IR index, histogram, and region-segmentation-based, and traditional machine learning vegetation detection, and Section II-C DL-CNN, and the application of DL to scene recognition.

TABLE I

O-D Camera Applications Summary

Method	Reference	Model	Approach
Localization	[1]	SLAM	Optical Flow and Particle Filter
Homography	[2-3]	Visual Geometry	3D View
Navigation	[4]	Sphere	Optical Flow
Calibration	[5]	O-D camera Calibration	Direct Spherical Calibration
Robotic Feature Tracking	[6]	Vertical Line Recognition	Robust Feature Descriptor

A. Omni-Direction Camera Setting

Intelligent perception is a critical element for autonomous applications in robotics. In order to improve the robot's perception, we used O-D sensors and reviewed the characteristics of O-D from different works in this section. The geometry of O-D camera is covered in various aspects [1]–[6]. Table I shows the related studies of O-D approaches. There are several approaches to localization. An approach for localization of UGV in a dynamic environment utilizes O-D vision and odometry [1]. Their approach couples optical flow with a particle filter and a database of former images mapped by the robot to establish the robot's global pose (x, y, θ).

The O-D geometry approach exploits the radial straight-line geometric feature of O-D vision to map to semantic primitives of the environment (doors, buildings, walls, edges, trees, corners, and radiators) and track these primitives as the robot moves through the environment. An O-D stereo and multiple images from different positions are used to extract the feature points for homography estimation in [2] and [3], where Lopez-Nicolas *et al.* [2] used multiple homographies from virtual image planes in O-D vision pairs. The authors generated a family of valid, virtual image plane homographies. The robot applies the matrix of resolved homographies to its control law to rotate the homography to the target, drive to the target, and then reorient to the target. Lui and Jarvis [3] use O-D vision homography on a moving platform to emulate stereo vision and extract multiple baselines to generate a map of the world. Lim and Barnes [4] explored the use of O-D sensors for navigation. Stone *et al.* [5] presented our approach for O-D camera calibration. Scaramuzza *et al.* [6] address identification of vertical line geometry in the image.

B. Visual and IR Index, Histogram, and Region Segmentation Based, and Traditional Machine Learning Vegetation Detection

The detection of surroundings is a significant application for navigation of autonomous systems to operate continuously in rough environments surrounded by vegetation such as trees, grass, and bushes. Tables II and III show the details of related works for vegetation detection and segmentation methodology.

TABLE II
VEGETATION DETECTION METHODOLOGY SUMMARY

Method	Reference	Model	Approach
NIR	[7-8]	Bayesian and Index Based	Satellite Remote Sensing
Information Extraction	[9]	Multispectral and Hyperspectral	Mixed Applications
NDVI	[10-11]	IR-Visual and Topographic Effects	Vegetation Indices

TABLE III
HISTOGRAM AND SEGMENTATION METHOD SUMMARY

Method	Reference	Model	Approach
NDVI	[12]	Laser - Visual Sensor Fusion	Lidar-NDVI Fusion
Histograms	[13-16]	Color - Gray Image and 3D	Region Matching
Region Segment	[17-20]	Hybrid Visual and IR	Semantic Map, Region and Edge

It is a critical task to identify vegetation in the robot's view for accurate navigation. The robot can easily navigate over grass or leafy objects whereas it is more difficult to move in a path consisting of tree trunks and constructed materials. The authors utilized satellite images for the detection of vegetation in multispectral of the near-infrared (NIR) spectrum and the visual spectrum in [7] and [8]. In order to use vegetation indices with mixed techniques, multispectral and hyperspectral data sets are employed in [9] and [10]. In addition to these approaches, large data sets were acquired from satellites for geographical classification. Bradley *et al.* [11] explored the application of the NDVI to vegetation perception in the DARPA Preceptor off-road UGV. NIR images are used as a threshold along with images from the visual sensor to detect the chlorophyll level corresponding to vegetation.

The histogram and segmentation approaches are summarized in Table III. Histogram and region segmentation approaches applied to vegetation detection [12]–[20]. A sensor fusion method applies the fusion of visual and point clouds from LIDAR [12]. References [13]–[16] apply histogram approaches, and references [17]–[20] apply semantic mapping and region-based segmentation approaches.

C. DL CNN Fusion of O-D IR and O-D Visual Stream

In Table IV, we can see the literature review for NN and DL methods. NN and machine learning techniques applied to obstacle avoidance are presented in [21]–[47].

A three-layer NN is trained to recognize red-colored visual objects [21], where the output of the NN is three movement commands (forward, turn right, or turn left). The method presented in [22] divides objects in the robot's environment into two map representations; first, a set of perceptual maps from each sensor derived by an NN feature extractor, and second, a self-organizing map algorithm with a spatial map representing the location of each of the objects in the perceptual maps. A growing cell structure NN approach builds up these map representations.

TABLE IV
NN AND DL METHOD SUMMARY

Method	Reference	Model	Approach
Neural Networks	[21-23]	Supervised Learning	Terrain Classification
Neural Networks	[24-27]	Unsupervised learning	Face Recognition
Autoencoder	[28-31]	Input Matching	Feature Extraction
Deep Fusion Network	[32]	Residual Network	
Classifier Fusion	[33]	Low and High Level Classifier	Classifier Fusion
CNN	[34-40]	Mixed	Road signs, road boundary
Databases	[41-42]	ImageNet City Scapes	Classification databases
Deep Learning	[43-47]	Various	Transfer learning
Machine Learning	[48-49]	SVM, RF, ANN	Comparison

A multilayer feedforward artificial neural network (ANN) [23] tracks dynamic obstacle motion by fusing ultrasonic and visual cues. The ANN is trained off-line using a relative error backpropagation algorithm. On-line the ANN predicts in real time the distance from the robot to the dynamic and stationary obstacles. The approach in [23] applies an environmental predictor ANN rather than a motion predictor to provide the robot with the areas occupied by obstacles. The algorithm fuses data from multiple sensors to identify the correlation between them and predicts the future sensor reading.

Dongshin *et al.* [24] explored the use of unsupervised learning for classification of terrain traversability. On-line learning is achieved by establishing a correspondence between the vehicle's navigation experience (successful traversals, slippage, collisions) using onboard sensors (Inertial Measurement Unit (IMU), bumper, and motor current), and visual characteristics of the terrain from a stereo vision sensor. The traversability level is established as affordance or rating of terrain difficulty to be traversed.

Chang and Kai-Tai [25] and Coates *et al.* [26] applied unsupervised feature learning to face recognition. The method used applies two simple learning algorithms: 1) a K-means clustering algorithm to prefilter the data and 2) agglomerative clustering to group simple cells into "complex cells" that are invariant features. Coates *et al.* [27] applied an analysis of the single-layer networks to unsupervised learning. Coates *et al.* [27] apply several unsupervised learning algorithms followed by a convolution classifier to evaluate the effect of different parameters used in unsupervised learning and CNN.

Autoencoders [28]–[31] are a class of NNs where the network is trying to approximate the identity matrix. The authors use the autoencoder as a feature extractor, where the learned weights of the sparse hidden layer represent features in the image. Rather than use labeled data, the autoencoder tries to match the output to the input and thus learn feature patterns in the data. Autoencoders [28] presented a method to convert high-dimensional data to low-dimensional data and to initialize weights enabling the deep autoencoder network to operate

TABLE V
DEEP FUSION NETWORK FOR O-D IR AND VISUAL STREAM

SECTIONS/ STEPS	DETAILS
III.A	O-D CAMERAS GEOMETRY AND SETTING
STEP 0	Camera calibration performed previously [5]
III.B	BASELINE METHODS
STEP 1	Apply the baseline MNDVI vegetation index and Thermal Region Fusion to extract vegetation regions for comparison.
STEP 2	Apply traditional machine learning approaches for comparison.
III.C	TRANSFER LEARNING AND FEATURE EXTRACTORS
STEP 3	Apply transfer learning with ImageNet and DL architectures along with Autoencoder Feature Extractor to the two streams to find the best network and parameters.
III.D	CNN FEATURE EXTRACTION AND DL TRADEOFFS
STEP 4	Apply transfer learning with ImageNet and DL architectures along with CNN Bottleneck Feature Extractor to the two streams to find the best network and parameters.
III.E	DEEP LEARNING TRADEOFFS AND DEEPFUSENET
STEP 5	Apply DeepFuseNet architecture to fuse the better of Step 3 or Step 4 O-D IR and Visual streams using the best architecture fed into a fully connected (Dense) CNN and softmax classifier to detect vegetation.
STEP 6	Compare Steps 1 - 5 results.

more efficiently. Hinton and Salakhutdinov [28] applied the above to recognition of character and facial feature.

A denoising autoencoder [29] applies intentional noise corruption of random pixels to a DL network to provide a more robust initialization of the network, thus guiding the intermediate steps based on the correction of the corruptions. Lee *et al.* [30] look at efficient sparse autoencoder techniques. Hao *et al.* [31] present a two-stream architecture, the first stream is a denoising autoencoder to encode pixel spectral values from the image and the second stream is the base image evaluated. A final CNN fuses the processed and base images.

Song *et al.* [32] applied residual learning to optimize CNN layer learning and to fuse the output of different hierarchical layers of the network. Halatci *et al.* [33] evaluated several low and high classifiers and looks at the fusion of the classification results for color, texture, and geometry for terrain classification. Zhuo *et al.* [34] applied automatically generated training data for CNN detection.

References [35]–[40] applied CNN to the classification of terrain and vegetation in the environment. In this work, we utilize the ImageNet [41] and City Scape [42] databases. We also reference [43]–[46] for refining our DL approach. Reference [47] is the authors' prior article, which compares a modified NDVI approach and our TRF approach for vegetation detection. Finally, Raczko and Zagajewski [48] compare support vector machines (SVMs), random forest (RF), and ANNs for classification of hyperspectral satellite images of ground cover. Omer *et al.* [49] compare SVM and ANN for mapping endangered tree species.

III. TECHNICAL APPROACH

The technical approach of our proposed work applies segmentation analysis on learning through our DeepFuseNet approach coupled with transfer learning from ImageNet to O-D visual and O-D IR images to provide vegetation detection, classification, and highlighting. The ultimate phase is the O-D vision system that utilized fusion of visual and IR images. Table V and Fig. 2 show the overall process.

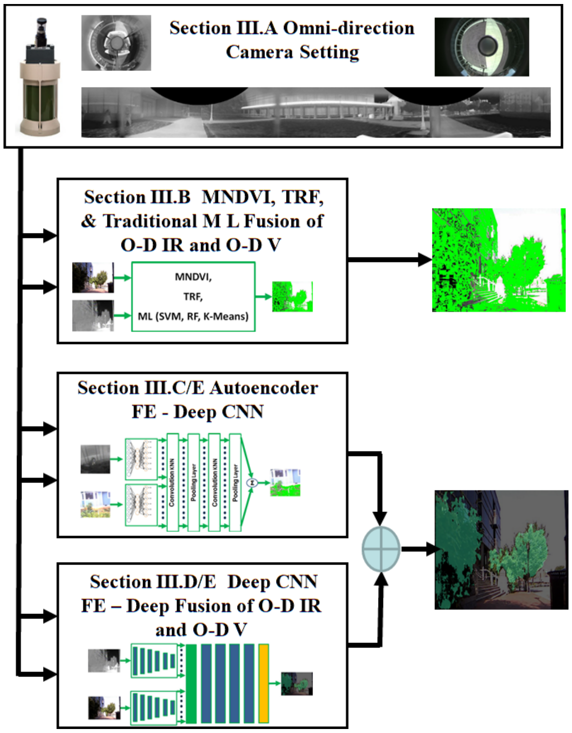

Section III.A Omni-direction Camera Setting
Section III.B MNDVI, TRF, & Traditional M L Fusion of O-D IR and O-D V
MNDVI,
TRF,
ML (SVM, RF, K-Means)
Section III.C/E Autoencoder FE - Deep CNN
Convolution KNN
Pooling Layer
Convolution KNN
Pooling Layer
Section III.D/E Deep CNN FE – Deep Fusion of O-D IR and O-D V

Fig. 2. Block diagram of the authors' technical approach.

The four subsections of Section III will cover the following.

1) Section III-A introduces the O-D sensor along with the O-D coordinate system setting.
2) Section III-B describes the baseline and traditional machine learning approaches.
3) Section III-C discusses the autoencoder feature extraction of the O-D IR thermal regions with the O-D visual (electro-optical) multispectral signature for the regions of interest from the sensor streams.
4) Section III-D discusses the CNN feature extraction and DL tradeoffs.
5) Section III-E discusses our DeepFuseNet architecture and the application of DL, transfer learning, and fine-tuning to solving the vegetation detection problem.

The DL fusion of visual and IR O-D data enables the robotic perception system to adapt to different lighting and environmental conditions. In this article, we will compare the baseline and traditional machine learning approaches to DL fusion approach shown in Fig. 2. The authors chose a two-stream merged fusion approach rather than a stacked image approach because the two individual inputs had different feature characteristics that we wanted to fully learn before merging the weights. We propose two architectures (Fig. 2) and compare them along with our previous baseline work, and other machine learning approaches.

A. Omni-Direction Camera Setting

In this work, the 360° field of view is obtained from the O-D imaging system utilizing spherical reflecting surfaces.

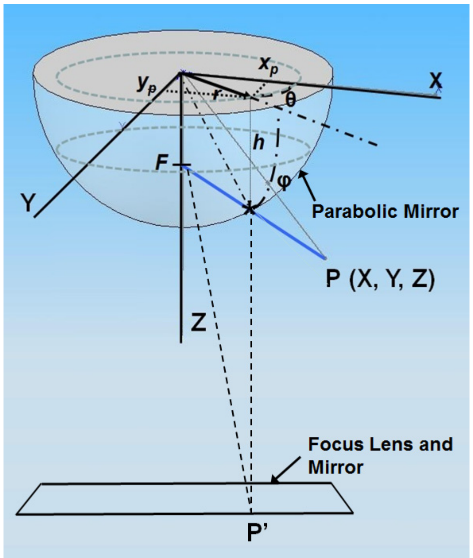


Fig. 3. Geometry of the O-D camera mirror system.

By means of these enhanced O-D systems, we generate the 360° image without any requirement of the image fusion of multiple sensors. Fig. 3 and the following equations describe the transformational relationship between coordinates of the image plane and coordinates of the spherical surface:

$$r^2 = x^2 + y^2 \tag{1}$$

$$\theta = \tan^{-1}(x/y) \tag{2}$$

$$\varphi = \tan^{-1}(r/z) \tag{3}$$

$$z = (r2 - h2)/2h \tag{4}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the pixel coordinates on the image plane, and $\boldsymbol{P}$ is the real-world point detected by the O-D sensor. The vertical distance from the mirror surface to the image plane is given by $\boldsymbol{z}$. The angles θ and φ define the angular direction of the ray from mirror's origin to the mirror coordinates of the detected pixel, and $\boldsymbol{r}$ is the horizontal distance from the mirror center to the coordinates of the observed point on the mirror surface. The orientation θ ranges from 0° to 360° around the edge of the mirror, and the pitch φ ranges from 0° when pointing straight down, to 90° when pointing at the horizon. The calibration parameter $\boldsymbol{h}$ is the height to the xy plane. Stone *et al.* [5] presented the calibration process for the O-D camera.

B. Baseline Methods—Index, TRF, and Traditional Machine Learning Approaches to Vegetation Detection

This section discusses the baseline methods, which we will compare to our DeepFuseNet approach. We are using low-cost O-D IR and O-D visual sensors to achieve better-fused vegetation classification. Fig. 4 shows the location of a point of interest in the O-D IR and O-D visual sensor. We show the unwrapped IR image and the extracted relevant section from the IR image. The blue arrows point to the relevant point in each of the images for comparison.

Stone *et al.* [47] evaluated the effectiveness of two approaches. First, we considered the classical index-based method of NDVI approach to vegetation detection, which looks at the red color bands in the visual spectrum and compares this to near IR spectrum. We use a modified normalized difference vegetation index (MNDVI), using FIR instead of NIR in the following equation:

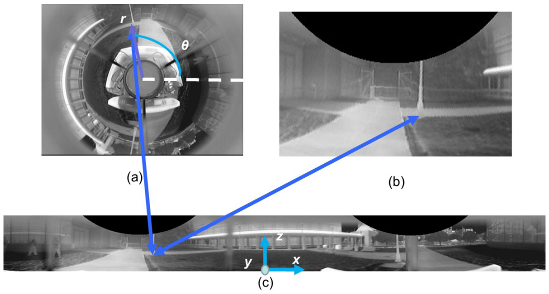


Fig. 4. Representative images from the O-D IR camera. (a) 360° O-D image with the r, θ geometry overlaid. (b) Cropped IR region of interest. (c) Unwrapped rectangular image with the x, y, z coordinate axis overlaid.

$$\text{MNDVI} = (I_{\text{IR}} - I_{\text{RED}})/(I_{\text{IR}} + I_{\text{RED}}). \tag{5}$$

The NDVI approach has a relatively high rate of false positives. Second, we considered a sensor fusion approach, merging the MNDVI and a thermal region segmentation approach. The NDVI approach works well in chlorophyll-rich vegetation but does not do as well in dry vegetation or desert scenes. The fused approach we demonstrated in [47] was to fuse the MNDVI (5) and thermal IR region (6)

$$\sigma_{\text{within}}(T) = \text{CDF}_{\text{B}}(T)\sigma_{\text{B}}^2(T) + \text{CDF}_{\text{F}}(T)\sigma_{\text{F}}^2(T) \tag{6}$$

where CDF_{B} (T) is the cumulative density function of the background below threshold T, $\sigma_{\mathbf{B}}^2(T)$ is the variance of the background histogram at T, $\text{CDF}_{\text{F}}(T)$ is the cumulative density function of the foreground above the threshold (T), and $\sigma_{\mathbf{F}}^2(T)$ is the variance of the background histogram at T. We then utilize the fused signatures to extract scene semantics from the O-D IR and visual images. In this work, we compare the results from the MNDVI red band approach, the fused IR region thermal segmentation approach, traditional machine learning approaches, and finally compare these methods to our two DL approaches presented in Sections III-C, III-D, and III-E.

One of the key issues with the current implementations of vegetation index-based processes is the high number of false positives that it produces. This method particularly struggles with synthetic materials that are green in color. The vegetation index approach has known failures in areas such as synthetic materials and paints that have high red absorption and behave similar to vegetation. Applying MNDVI index-based vegetation detection alone has a high incidence of false positives.

These issues with false positive detects are in the areas where the bands overlap when using just the MNDVI approach. Stone *et al.* [47] used the MNDVI approach on our data and we show the results in Fig. 5. The effect of this approach in this example is a relatively large number of false detects. The average false detects was on the order of 31%.

Human vision system is adapted to using color and texture to distinguish objects. This distinguishing process is accomplished by using the important indicators of objects, the color,

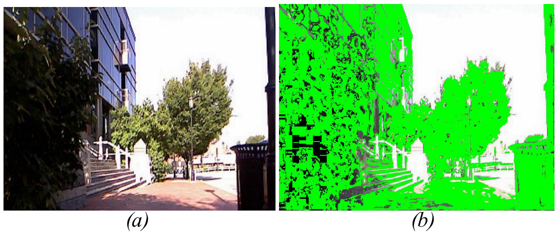


Fig. 5. MNDVI approach. (a) Original image. (b) Processed image using MNDVI vegetation index approach. This result has a high number of false detects.

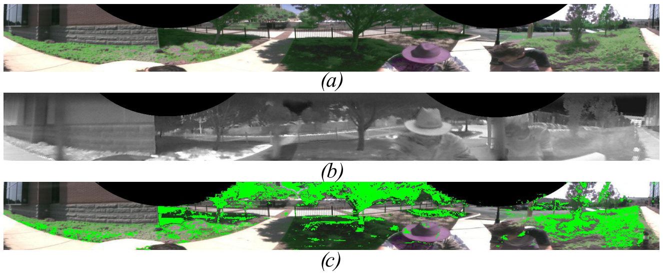


Fig. 6. MNDVI/TRF fusion results. (a) Visual O-D image. (b) IR O-D image. (c) TRF fusion results.

and reflectivity. The baseline methods do not adequately capture these important features, which is why we need DL to learn these features in a similar fashion to the human perception model.

Fig. 6 shows the results of the MNDVI/TRF fusion method, visually showing the relationships of the results. While the TRF reduces the false positives, it does not completely capture the vegetation region. This is why we look at DL to learn the feature patterns in the data.

In [47], we address the false detection problem in the index-based approach, using semantic extraction based on thermal regions to emphasize on vegetation detection in order to solve the detection problem of the false positives. The material type will determine the region characteristics in the IR images and will cluster these regions. When the lighting or temperature conditions alter, any change in the color of the objects will not change the thermal view of the clusters in the IR images. By using DL to recognize and learn this region-based color signature and to learn the thermal behavior, we can merge the features learned by the two networks. By utilizing the characteristics of soil and vegetation, day time images have darker vegetation regions in thermal images since the soil is warmer than the vegetation. At night, vegetation and soil reverse and the vegetation is lighter. When dust or fog obscure the visual image, the vegetation region is still visible in the IR image. These attributes are an advantage of applying DL to both the visual and the thermal information.

We also looked at a few traditional machine-learning approaches for comparison. They were SVMs, RFs, and K-means. SVMs are a class of supervised learning models or classifiers that separate the data into clusters of points in space with a hyperplane separating classes. Initially, the SVM method required classes to be linearly separable. If the data are not linearly separable, we use a nonlinear kernel to separate the data by moving it into a higher dimensionality.

RFs are an ensemble learning or classification approach that generates a forest of decision trees. By randomly generating multiple decision trees, we obtain better accuracy and reduce overfitting to the training data. RF improves accuracy by randomly aggregating multiple decision trees into one big meta-classifier or RF. The aggregate classifier then averages the votes from each of the individual classifier models injecting randomness in the training data and random feature vector selection. Finally, K-means is an unsupervised learning approach where we cluster the data into unique groups of similar features. The approach named K-means gets its name because the algorithm groups the data into K unique clusters where the center of the cluster is the mean of the values in the cluster.

The contribution of this work is a computational foundation for a new approach to multispectral sensor fusion, which identifies semantically significant object classes using DL sensor fusion of O-D visual and O-D IR sensors.

We will evaluate our two DL architectures against five baseline methods: index-based (MNDVI), thermal region-based fusion, and the three traditional machine learning approaches. Our two new architectures are as follows:

1) Two stream sparse autoencoder feature extractor (SAFE) fused by CNN;
2) DeepFuseNet—two Conv NN feature extractors, which apply transfer learning, merged into a final output deep fusion network.

C. Autoencoder—CNN

In the first architecture, we input the two streams into two SAFEs in order to extract salient features from each media type before fusing them with a deep CNN backend. This approach allows the independent extraction of key thermal, color, and texture features before fusion of the IR and visual content. This approach facilitates a more robust extraction of the regions of vegetation and other materials.

Our algorithm uses the output of the two SAFEs as the input of the deep CNN for fusion/classification. We then compare the SAFE-CNN architecture against the baseline index-based, thermal region-based, and machine-learning methods to vegetation detection.

The SAFE-CNN architecture feeds the visual and IR input streams into two respective autoencoders that then feed forward the input images through the network to minimize the difference between the input and output, thus learning the feature properties of the materials in the scene. The final CNN fuses these two autoencoder feature outputs. We build the autoencoder of a network of basic “Neuron” nodes (Fig. 7) which applies unsupervised learning to approximate the kernel feature map of the scene.

Fig. 7 represents one of the autoencoder NNs, using an unsupervised learning approach by applying backpropagation of the error to match the output to the input. We start with a set of unlabeled training data $\{\boldsymbol{x1}, \boldsymbol{x2}, \boldsymbol{x3}, \boldsymbol{xn}\}$ where $\boldsymbol{x_i \mathcal{E} R^n}$ and the network learns by matching the output to the input, or by setting $\boldsymbol{y_i} = \boldsymbol{x_i}$. As the weights, biases, and activations are set to minimize the error between the two, the hidden feature layers will learn patterns in the image set. The auto-encoder

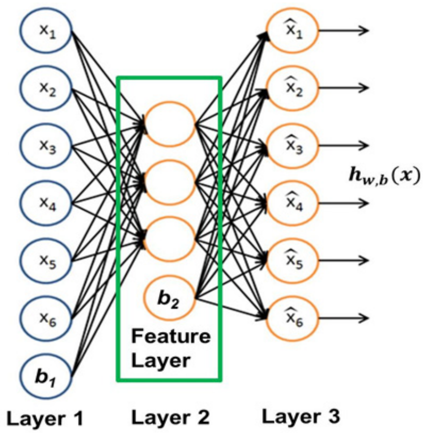


Fig. 7. Three-layer NN with a hidden layer that is sparse or constrained, which forces the network to learn certain features in making the output match the input. This is an autoencoder.

will try to learn the output function $\boldsymbol{h}$, such that $\boldsymbol{h_{Wb}(x)} \approx \boldsymbol{x}$’ is satisfied to make the output match the input, which means that it is trying to estimate the identity function.

Putting constraints on the autoencoder NN forces the autoencoder to learn the structure (patterns) in the data. For instance, with some of the color and texture features of grass correlated, the Autoencoder detects this pattern.

There are several ways to put constraints on the network. One is to limit the number of units in the hidden layer. Another is to place a sparsity constraint on the hidden units in the network. With a sparsity constraint, the auto-encoder will still be able to detect interesting structures in the data even if the number of hidden units is large. If we use a sigmoid function

$$a_i^l = \frac{1}{(1+e^{-x})} \tag{7}$$

as our activation parameter, we can think of a given feature as being active if its activation for feature $\boldsymbol{i}$ at layer $\boldsymbol{l}$, $\boldsymbol{a_i^l}$ is close to $\mathbf{1}$ and inactive if it is close to $\mathbf{0}$. We want each feature node in the hidden layer to be inactive most of the time. We want that location to activate only when there is sufficient correlation in the data. Therefore, the algorithm puts a constraint on the cost function that drives this condition. We add this sparsity constraint parameter to our cost function to place a penalty on a neuron activating when the image region lacks correlation to the region of interest. This will be the case if the feature is not present. The activation $\boldsymbol{a_i^l}$ represents the activation of the hidden node at the $\boldsymbol{l}$th layer for a given input $\boldsymbol{x(i)}$, as shown in the following equation:

$$\hat{p}_j = \frac{1}{m}\sum_{i=1}^{m} a_j^l x^{(i)}. \tag{8}$$

We then enforce the constraint that $\boldsymbol{\hat{p}_j = \rho}$, where $\boldsymbol{\rho}$ is the sparsity parameter. We constrain the activation function to be small and close to zero, such that the output matches the input within the convergence error. Our cost function will then be

$$J(W,b;x,y)=\frac{1}{2}\|h_{W,b}(x)-y\|^2. \tag{9}$$

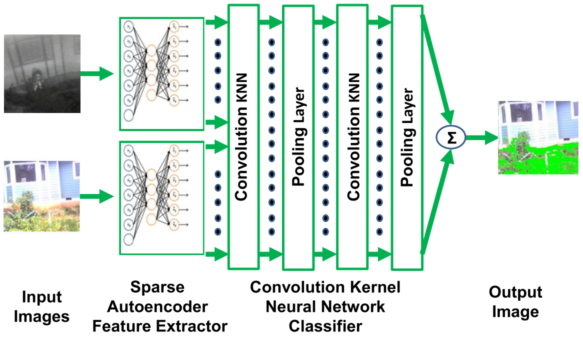


Fig. 8. First fusion architecture with two autoencoders feeding into a two-layer CNN with max pooling layers.

We then apply the L1-regularization (10) constraint to penalize $\boldsymbol{\hat{p}_j}$ the farther it diverges from zero

$$L_1 = \lambda\sum_{j=1}^{n_l}\omega_j(\rho||\hat{\rho}). \tag{10}$$

This will force $\boldsymbol{\hat{p}_j \approx \rho}$ and our cost function becomes

$$J_{\text{sparse}}(W,b) = J(W,b) + \lambda\sum_{j=1}^{n_l}\omega_j(\rho||\hat{\rho}). \tag{11}$$

To learn the weights for the nodes in the network, we conduct a forward pass with initial weights, and then use backpropagation to refine the weights to match the output to the input streams from the O-D IR and visual cameras. We use backpropagation to adjust the weights as follows:

$$\delta_j^l = \left(\left(\sum_{j=1}^{n_l} W_{ji}^l\delta_j^{l+1}\right) + \lambda\sum_{j=1}^{n_l}\omega_j(\rho||\hat{\rho})\right) f\left(\acute{z}_i^l\right). \tag{12}$$

Due to the $\boldsymbol{L_1}$ sparse constraint on the network, as the network converges to the optimum it learns the new weighting on the hidden layers to match the output to the input. Since we have placed the $\boldsymbol{L_1}$-regularization constraint on the network, the algorithm suppresses less important features. The network will learn the features in the data, and key structures in the two input streams will emerge. We feed these key structures into a pretrained CNN to fuse the features. Finally, we feed the two autoencoder outputs into the final deep CNN to fuse their features and provide a fused sensor output as shown in Fig. 8.

D. Deep-CNN Feature Extraction

Our second algorithm applies two-stream multilayer kernel filter deep CNN feature extractors to extract salient thermal, color, and texture features from the two different visual and IR input streams. We then use the final DeepFuseNet to merge the feature-vector weights.

The deep CNN will consist of a sequence of two sets: a convolution (conv) kernel layer, a rectified linear nonlinearity (ReLU) layer, and then a max-pooling layer. At each conv layer, the following equation denotes the feature map from the previous layer $\sum_{i=0}^{m_1^{l-1}} Y_i^{l-1}$ convolved with the learnable kernel filter K_{ij}^l for the current layer. The result becomes the input to the activation function to form the ith feature map in layer $\boldsymbol{l}$ of Y_j^l

$$Y_j^l = f\left(B_j^l + \sum_{i=0}^{m_1^{l-1}} Y_i^{l-1} * K_{ij}^l\right) \tag{13}$$

where the output at layer $\boldsymbol{l}$, $\boldsymbol{Y_j^l}$, is a function of the matrix $\boldsymbol{B_j^l}$, which is the matrix of bias at each layer $\boldsymbol{l}$ added to the sum of the spatial conv of the output of layer $\boldsymbol{l-1}$, $\boldsymbol{Y_i^{l-1}}$ with the matrix of Kernel, $\boldsymbol{K_{ij}^l}$ at layer $\boldsymbol{l}$ over an $\boldsymbol{i \times j}$ window.

The following equation gives the size of the output feature map:

$$M_x^l = \frac{M_x^{l-1} - K_x^l}{S_x^l + 1} + 1; \quad M_y^l = \frac{M_y^{l-1} - K_y^l}{S_y^l + 1} + 1 \tag{14}$$

where each layer has $\boldsymbol{M}$ maps of equal size $(\boldsymbol{M_x}, \boldsymbol{M_y})$. The kernel filter shifts over the image such that it does not go outside the image. The kernel is of size $(\boldsymbol{K_x}, \boldsymbol{K_y})$, where the index $\boldsymbol{l}$ indicates the layer number, and each map in layer $\boldsymbol{L^l}$ is connected to a subset of maps in the previous layer $\boldsymbol{L^{(l-1)}}$.

The max-pooling layer is a down-sampling layer that applies a $\boldsymbol{k \times k}$ pooling window to reduce the dimensionality of the image. It also improves the spatial invariance by reducing the resolution of the image. In our case, we take the maximum value and apply it to the $\boldsymbol{k \ \times k}$ kernel patch by

$$m_j = \max_{K \times K}\left(m_i^{kxk} u(k,k)\right) \tag{15}$$

where each pixel in the new window is assigned the value $\boldsymbol{m_j}$, which is the maximum value of the search of window $\boldsymbol{u(k \times k)}$.

Each of the kernel layers provides a refinement of the image representation, with each layer being more invariant than the previous. These kernel layers define the feature maps of the image. Each of these kernel-image feature maps has normalized coordinates, both in the image and in the Hilbert space $(\boldsymbol{H})$. The higher dimensional kernel-feature map is the dot product of similar layer features. Applying the multilayer convolutional kernel to the image-feature map results in patches of like features and the fusion of the kernel prefilters. This represents a spatial convolution (conv) over the image. The resulting semantic feature vector is the input to the 2-D vegetation classification algorithm.

The following parameters characterize each conv layer: size and number of maps, the kernel window size, the stride (step over or skipping factor), and the connection strategy. The conv layers of our network have the following key parameters in line with the findings of [27]. We further define these parameters below.

Stride—The spacing between patches for feature extraction, or in other words, the number of pixels skipped before again applying the receptive field. The results show the best performance at a stride of $\boldsymbol{S = 1}$, with a clear downward trend in performance as step size increases. This is a tradeoff between accuracy and computation time.

Kernel Size—The size of the window or field over which feature extraction is applied. Coates *et al.* [27] used 6, 8, and 12 pixels, with generally 6 pixels working the best. In our experiments, we use 2, 3, and 6 pixels, again with 6 pixels working the best.

Number of Features—Coates *et al.* [27] evaluated 100, 200, 400, 800, 1200, and 1600 learned features. On average, the algorithms performed better by learning more features, although the increase above 800 was gradual.

Image Registration—The matching of features in the two images to align them. In this work, the DL algorithm itself accomplished this. We apply DL directly to learn the key features and the resulting geometric transformation (homography) to align the two images.

We conducted a study looking at the performance of various model architectures and parameters to define the optimum combination for our research. Architectures evaluated were AlexNet, SqueezeNet, ResNet50, VGG16, ResNet101, and ResNet152. Parameters evaluated were learning rate (LR) $\boldsymbol{\eta}$ and momentum $\boldsymbol{m}$. LR $\boldsymbol{\eta}$ is the rate at which the network learns and updates the network weights. The following equation shows the weight update process:

$$w \overset{\text{update}}{\longleftarrow} w - \eta \frac{\partial L(x, y)}{\partial w} \tag{16}$$

where $((\partial L(x, y))/\partial w)$ is the backpropagation method of updating the weights in the network by comparing the actual output to the desired output with the partial derivative of the error with respect to the weights, and $\boldsymbol{L(x, y)}$ is the cross-entropy loss function given by

$$L(x, y) = -\left[y \log x + (1 - y) \log (1 - x)\right]. \tag{17}$$

We evaluated LRs of 1e-1, 1e-2, 1e-3, 1e-4, and 1e-5, adjusting the LRs to minimize overfitting.

Momentum $\boldsymbol{m}$—helps to drive the algorithm through a local minimum to find the true minimum. It is similar to physical momentum, but in DL, the momentum is resisting the gradient descent update to the training weights near a local minimum. We evaluated momentum values of 0.9, 0.8, 0.75, and 0.7 when evaluating the above architectures that we trained on ImageNet. We also applied Dropout $\boldsymbol{d}$ to reduce overfitting by randomly dropping out neurons in the network at probability $\boldsymbol{p}$.

E. DL Fusion Network (DeepFuseNet)

Our architecture (layer level depiction Fig. 9) for the DL fusion network (DeepFuseNet) uses two bottleneck feature extractors (BFEs) fed into two DL VGG 16-layer conv networks, one for the visual stream and one for the IR stream, trained on ImageNet, with 1 million images and 1000 classes. We then apply transfer learning by freezing the lower 15 layers and training a new fully connected top layer, the new head model, trained on our smaller vegetation data set. The transfer learning method leverages the patterns learned from ImageNet and refines the learned weights in the top layer using our vegetation-specific training data.

Each layer will perform conv, followed by ReLU nonlinear layer and a max-pooling layer with a 2 × 2 window and down sample. We conducted a series of experiments to fine-tune the parameters of the new top of the VGG feature extracting networks before concatenating into the final top

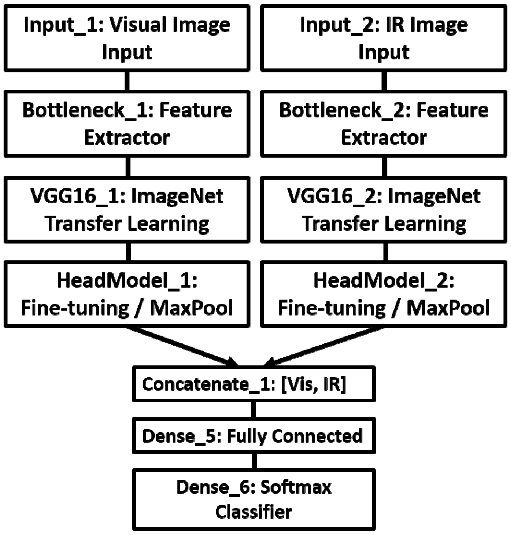

Input_1: Visual Image Input
Input_2: IR Image Input
Bottleneck_1: Feature Extractor
Bottleneck_2: Feature Extractor
VGG16_1: ImageNet Transfer Learning
VGG16_2: ImageNet Transfer Learning
HeadModel_1: Fine-tuning / MaxPool
HeadModel_2: Fine-tuning / MaxPool
Concatenate_1: [Vis, IR]
Dense_5: Fully Connected
Dense_6: Softmax Classifier

Fig. 9. Model block diagram with two VGG16 inputs with visual and IR input.

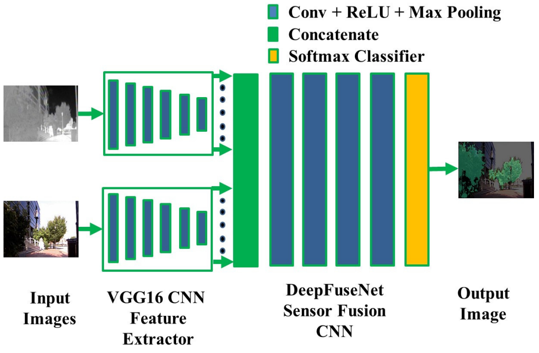

Conv + ReLU + Max Pooling
Concatenate
Softmax Classifier
Input Images
VGG16 CNN Feature Extractor
DeepFuseNet Sensor Fusion CNN
Output Image

Fig. 10. DeepFuseNet fusion architecture with two CNN feature extractors feeding into a seven-layer CNN with max pooling and a classification layer.

two dense and softmax classifier layers. Fig. 9 presents the layer architecture for the two parallel VGG 16 models with five convolutional/max pooling blocks and the concatenated input into the final two dense layers. The two concatenated feature extraction networks feed into a top network of two dense convolution layers and a softmax classifier.

Fig. 10 represents the concept-level depiction of the DeepFuseNet. Fig. 10 shows the visual and IR image streams into a feature extractor and then feeding the DeepFuseNet. The output of the network feeds into a softmax classifier to output the image with the vegetation areas semantically detected and classified.

We utilized network-filter visualization techniques [41] to highlight the features that activate the network activation filters. Fig. 11 presents the conv filter maps for the first conv layer and three representative channels. Fig. 11 shows the original image, channel 3, and channel 13 activating on different features in the image. We can see from Fig. 11 that the various activation filters are learning (activating) on different feature sets. Fig. 11(a) shows the original visual image. In Fig. 11(b), the filter is learning vertical edges in the vegetation; in Fig. 11(c), the filter is learning primarily the sky and road surface, which are both a gray color in this example.

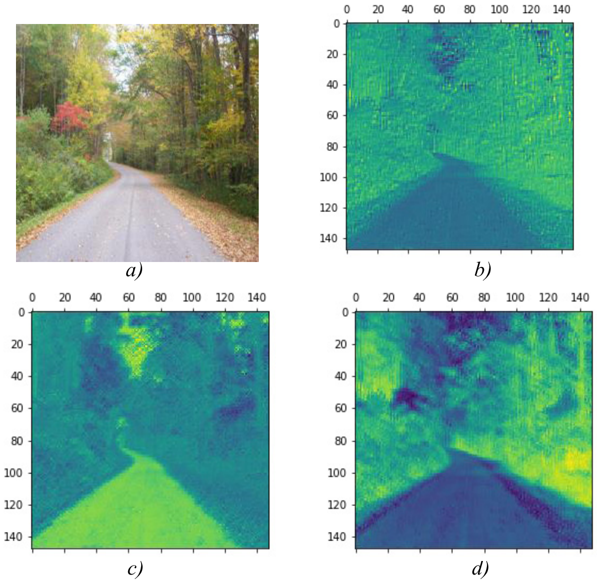


Fig. 11. First conv layer feature maps for three out of the 32 channels. (a) Original image. (b) Feature map for layer 1 channel 3. (c) Feature map for layer 1 channel 13. (d) Feature map for layer 1 channel 19.

Finally, in Fig. 11(d), the filter is learning tree trunks and splotches of leaves.

IV. Experiments

The experimental results are organized into five sections. Section IV-A gives details of the O-D camera data setting. In Section IV-B, the results of the baseline methods, FIR/visual modified vegetation index, thermal region, and traditional machine learning approaches are described. Section IV-C describes the results of the autoencoder feature extractor-CNN fusion of O-D IR and visual streams. Section IV-D describes the results of the authors' deep CNN trade studies. Section IV-E describes the results of our bottleneck CNN feature extractor/DL fusion network, which is the preferred method of O-D IR and visual stream fusion using our DeepFuseNet approach.

A. Omni-Direction Camera Data Setting

We have five data sets used in this work. The first two data sets 1 and 2 captured images from the O-D FIR camera and Kinect visual camera systems with 38 images each; we later captured additional IR and visual data from our O-D IR and O-D visual cameras, data set 3 with 13 730 images. Data set 4 is the ImageNet data set consisting of 1 231 167 images and over 1000 classes. Data set 5 consists of images of vegetation, trees, and grass extracted from the internet using a web scraper with 900 images in each of three classes. We summarize the data sets in Table VI. The ImageNet data set had 48 238 validation images, and 50 000 test images out of 1 231 167 training images. Our vegetation training set includes 280 validation, and 200 test images.

We process the resulting data first with the three baseline approaches: the MNDVI, the infrared thermal-based region segmentation (ITRS), and traditional machine learning. We then compare the results to the authors' two approaches:

TABLE VI
CAMERA DATA SETTINGS

Data Set	Image Type	Size	Sensor
1	Far-infrared	38 images	O-D IR
2	Visual Kinect	38 images	Kinect
3	O-D V & IR VCU Campus	13,730 images 13,730 images	O-D Visual & O-D IR
4	ImageNet	1,231,167 Images, 1000 Classes	Visual & IR
5	Web Scraper Grass, Tree, Veg	900 images 3 Classes	Visual & IR

TABLE VII
BASELINE AND SENSOR FUSION RESULTS

Attribute	ACC	Loss	Val ACC	Val Loss	% False Positive
MNDVI	86.74	N/A	N/A	N/A	32.5
TRF	75.16	N/A	N/A	N/A	11.5
SVM [48]	68.0	N/A	N/A	N/A	N/A
SVM [49]	70.7	N/A	N/A	N/A	31.6
RF [48]	62.0	N/A	N/A	N/A	N/A
ANN[48]	77.0	N/A	N/A	N/A	N/A
ANN[49]	69.7	N/A	N/A	N/A	32.8
SVM*	64.4	N/A	N/A	N/A	N/A
RF*	57.0	N/A	N/A	N/A	N/A
K-Means Cluster*	62.7	N/A	N/A	N/A	N/A
SAFE-CNN	86.0	0.32	79.0	0.46	-----
DeepFuseNet	95.6	0.1	92	0.2	1-2

* The authors' traditional machine learning results

the autoencoder feature extractor-CNN fusion network and the deep transfer learning fusion network (DeepFuseNet). To address the issue of training on our limited data set, we apply transfer learning from the larger generic ImageNet data set. Our approach also applies semantic learning to extract the vegetation features in the data.

B. Baseline Visual and IR MNDVI, TRF, and Machine Learning Vegetation Detection

The results from the baseline index-based MNDVI, TRF, and traditional machine learning-based approaches are presented in this section.

Stone *et al.* [47] demonstrated that the accuracy of the MNDVI approach is obtained as 85.6% for the true positives and 32.5% for the false positives. The accuracy of the TRF approach is 75.16% for the true positives while the false positives are obtained at 11.75%.

We also compare our methods to a few traditional machine-learning approaches. We compare to results from [48] and [49] and to ML experiments with our data on our own implementation of SVM, RF, and K-means clustering.

Table VII, rows 3–7 present results from [48] and [49] for SVM, RF, and ANN. The authors' implementation of SVM*, RF*, and K-means cluster* are presented in Table VII, rows 8–10. SVM achieves accuracies of 68.0% and 70.7% in [48] and [49] and a false positive rate of 31.6%. SVM* achieves an accuracy of 64.4%. RF achieves an accuracy of 62% in [48] and RF* an accuracy of 57%. ANN achieves an accuracy of 77% and 69.7% in [48] and [49]. ANN [49] had a false positive rate of 32.8%. The others did not present the false positive rate. K-means achieves an accuracy of 68% in [48], 70.7% in [49], and K-means clustering* achieves an accuracy of 62.7% with a false positive rate of 46.3%. (Note: * The authors' traditional machine learning results.)

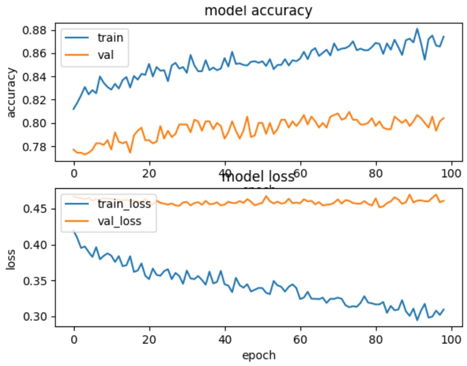


Fig. 12. Results of applying transfer learning and fine-tuned on our vegetation data set in order to extract vegetation features. Fusion model trained with autoencoder input. Final accuracy 79% and loss 46%. Results show the method does not generalize well.

The low accuracy and high false positives from these methods highlight the need for a new fusion method, which is why we moved to the deep transfer learning approach. We summarize the results of these baseline methods and our two methods in Section IV-E (see Table VII). We conclude from these results that we need a more robust method to detect the vegetation.

C. Autoencoder—CNN Fusion of O-D IR and Visual Stream

The SAFE applies an autoencoder and bottleneck CNN to extract salient features from the image. We achieve nearly 99% accuracy and low loss when trained on the ImageNet data set, but the network does not generalize well when applied to our data.

The unsupervised learning SAFE-CNN autoencoder approach captured the features well when trained on ImageNet, but it was not as effective when trained with our data on the fusion network. Fig. 12 shows the training and validation results, after applying transfer learning and fine-tuning on the authors' data set. The SAFE-CNN only achieves a training accuracy of 86% and a validation accuracy of about 79%. We will continue to explore why this occurred and how we can fine-tune the model to better utilize this approach.

We hypothesized that the positive impact of using unsupervised learning with two sparse autoencoder neural network feature extractors into a CNN would be that we would more effectively capture the vegetation pattern. However, we did not achieve this. Once successful, we believe this will provide better performance and less false positives.

D. Deep CNN Trade Studies

Since the authors did not have a large data set from which to train, we applied transfer learning to the problem. We utilized the large ImageNet [42] data set with over 1.2 million images and 1000 classes to get an initial trained model and then applied transfer learning and fine-tuning to refine the model to our smaller data set. The transfer learning process freezes the weights of the lower layers of the network trained on the

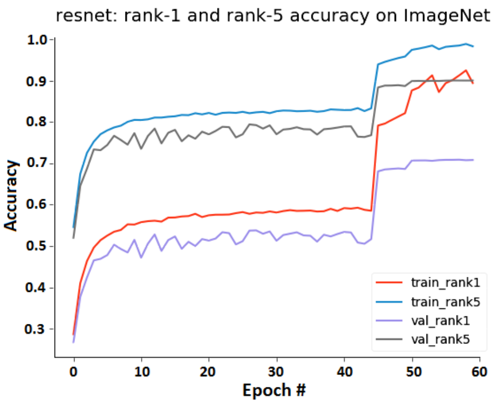


Fig. 13. Training loss and accuracy for ResNet trained on ImageNet data set.

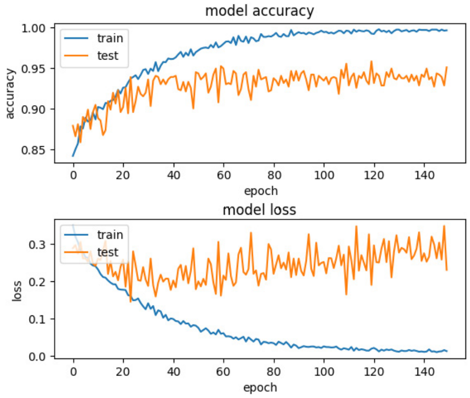


Fig. 14. Training loss and accuracy for VGG16-based BFE leveraging ImageNet transfer learning and fine-tune trained on our vegetation data set.

larger ImageNet and adds a new top fully connected section; we then trained the top layers of the network on our smaller data set. This allows the network to retain the learned features of the lower activation filters, and to refine the learning in the upper layers based on the new training data.

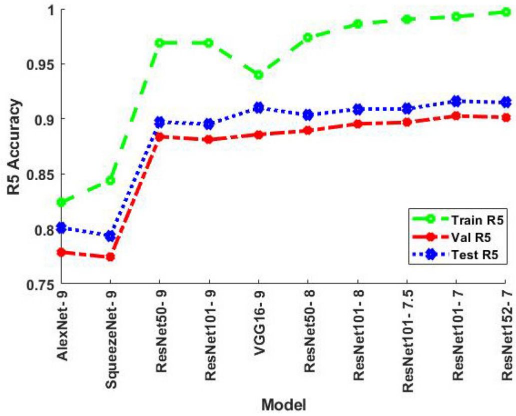


Fig. 15. Rank 5 accuracy across the various models with different momentum. Again, there were slight improvements for the deeper models at momentum = 0.75 and 0.70.

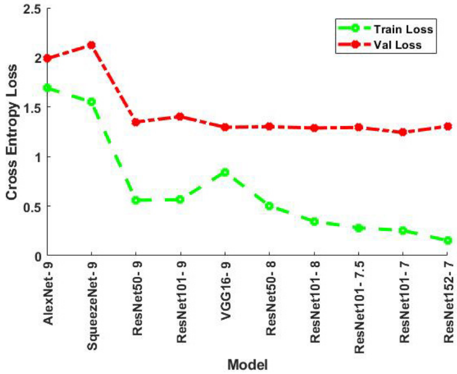


Fig. 16. Training and validation cross-entropy loss across the various models with different momentum. A test curve is not presented since test was a prediction from an image and did not have a loss.

Based on our literature search, our initial hypothesis was that the best learning would come from the use of the deeper networks of ResNet50, ResNet101, or ResNet152. However, after our trade studies comparison of these various methods trained on ImageNet, we saw that ResNet50, ResNet101, ResNet152, and VGG16 all had similar performance, with VGG16 having less overfitting. As a result, we chose VGG16 as the DL primary network model that we modified for application of transfer learning and bottleneck feature extraction. Fig. 13 is a representative ResNet model trained on ImageNet showing the accuracy improvement at Epoch # 45 after adjusting the LR for fine-tuning. We similarly evaluated the other models to determine the best architecture for our research. Fig. 14 shows results for the BFE CNN for tuning the starting weights of our process. The BFE does not perform as well as the SAFE on initial ImageNet learning. The BFE-CNN only achieves 94%–95% feature extraction compared to the 99% achieved with SAFE. However, the BFE-CNN with DeepFuseNet training achieves better generalization than the SAFE-CNN fused model. As a result, the BFE-CNN-DeepFuseNet applied to our data performs better than the SAFE-CNN fusion model with higher classification accuracy. Fig. 14 shows the training/test accuracy and loss for the VGG16-based BFE. The trade study comparison results for the AlexNet, SqueezeNet, ResNet50, ResNet101, VGG16, and ResNet 152 models evaluated are presented in Figs. 15 and 16. We trained several models to find the best performance for our transfer learning experiment.

For the initial assessments, we held the LR the same (1e-3) and compared all the models at the same momentum = 0.9. Our findings were that AlexNet and SqueezeNet had lower performance than the other models, and ResNet50, ResNet101, ResNet152, and VGG16 had comparable results. We then looked at the deeper ResNet models and varied the momentum to see if that would improve the accuracy. There were slight increases but not significant enough to accept the increased processing requirements.

Fig. 15 is the Rank 5 accuracy. The Rank 5 accuracy is the percentage of the assessments that the target is in the top five highest probability predictions. Again, the results were repeatable with AlexNet and SqueezeNet being lower at about 80% and the accuracy of the others ranging from about 90%–92%. Fig. 16 presents the training and validation cross-entropy loss for the models evaluated. Similarly, other than AlexNet and SqueezeNet, the performance for the deeper models is very consistent.

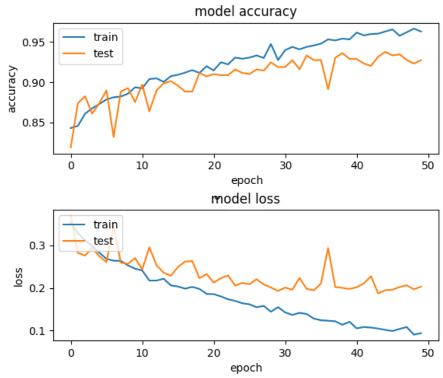


Fig. 17. Fine-tuned merged model trained on vegetation data.

From this series of fine-tuning experiments with AlexNet, SqueezeNet, VGG16, ResNet50, ResNet101, and ResNet152, we selected VGG16 for our transfer learning and fine-tuning adaptation, as it had reasonable performance and less overfitting without the computing cost of the deeper networks. We initially trained the models on ImageNet data and then did transfer learning to train on our smaller data set.

E. DeepFuseNet

For our DeepFuseNet approach, we merged two transfer learning VGG16 BFE-CNN models, one for the visual image stream and one for the IR image stream. We chose a two-stream approach rather than a stacked band approach to allow the two BFE to extract the different characteristics unique to the stream type. For instance, color and texture from the visual stream and thermal signature regions from the IR stream. The outputs of the two models were concatenated and fed into our DeepFuseNet with two fully connected dense layers and a softmax classifier to highlight the semantic region of interest.

Fig. 17 shows the training results for the merged DeepFuseNet model. The training accuracy achieves 95% in 50 epochs. The validation test accuracy is about 92%.

Fig. 18 shows representative samples of vegetation recognition with the DeepFuseNet. The capture accuracy is high at 95.6%. We reduced false positives to 1%–2% compared to the baseline approaches.

Table VII shows the comparison between the baseline MNDVI, TRF, the traditional machine learning, the SAFE-CNN, and the DeepFuseNet approaches. Table VII, row 1 shows the visual MNDVI vegetation detection approach. Row 2 is the ITRS or TRF approach. We present the traditional machine learning results in rows 3–10, and finally the SAFE-CNN and DeepFuseNet results in the last two rows of Table VII.

The authors found the DeepFuseNet fusion approach to have both the best accuracy and relationship between true positives and false positives. The DeepFuseNet has the best accuracy with 95.6% true positives and reducing the false positives to less than 2%, which is a 16× improvement over the index based alone. The reason for this improvement in false positives is the ability of the DL and feature fusion approach to learn the key features in the data. The DeepFuseNet approach utilized O-D FIR and visual sensors to cover wider view rather than limited view in the traditional vision systems. In terms of computational efficiency, our DeepFuseNet approach, with the ability of processing wider area, outperforms the approaches that require multiple iterations or vision sensors for the same real-time robotic applications.

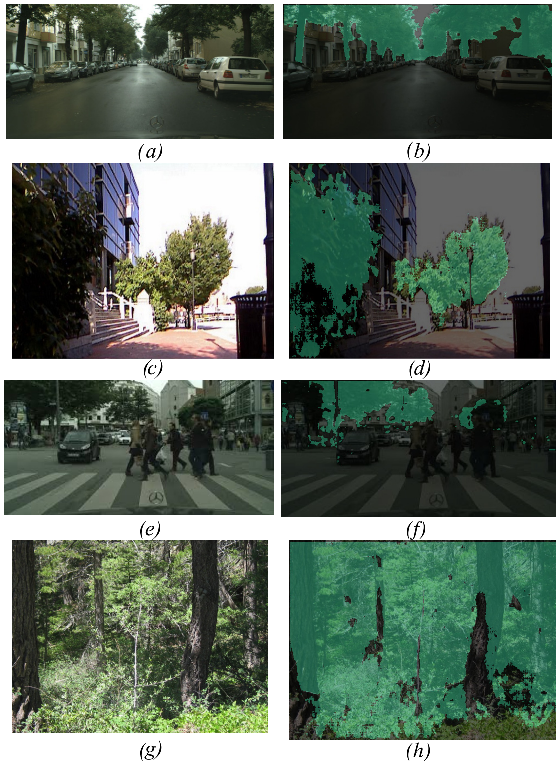

(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 18. Vegetation detection results. (a) Tree lined street image 1. (b) Vegetation Mask 1. (c) Virginia Commonwealth University (VCU) entrance image 2. (d) Vegetation Mask 2. (e) City scape image 3. (f) Vegetation Mask 3. (g) Forest image 4. (h) Vegetation Mask 4.

V. Conclusion

It is clear that the DeepFuseNet approach provides the best overall results, with the best generalization from ImageNet data set to our smaller vegetation data set. The SAFE and CNN Fusion approach did not give as good a result as we expected because it did not generalize well from ImageNet to our small data set. Results for the baseline methods of the MNDVI approach, TRF, and traditional machine learning approaches ranged from 57% to 86% accuracy with high false positives.

The authors' DeepFuseNet approach demonstrated a 95.6% accuracy for true positives and a 93.8% reduction in false positives. This improvement is due to the deep layers of feature encoding that employ a large spatial context of the network for learning the salient characteristics of the vegetation and nonvegetation features.

We will apply this DeepFuseNet approach to our Pioneer robot platform for follow-on experiments. Future studies will refine the DeepFuseNet method, and further explore the SAFE-CNN approach. We will augment our DeepFuseNet approach with texture analysis and context-based reasoning algorithms to better distinguish surrounding in a spatial scene that contain objects such as grass, trees, and bushes. We will apply this approach to future robot platforms in urban and rural environments for object detection classification and tracking for autonomous navigation in tough conditions.

Acknowledgment

The authors wish to thank the Office of Naval Research (ONR), Arlington, VA, USA, the Naval Surface Warfare Center Dahlgren Division (NSWCDD), Dahlgren, VA, USA, and the Marine Corps Warfighting Lab (MCWL), Quantico, VA, USA, for their support of this study. They would also like to acknowledge Adrian Rosebrock and his PyImageSearch Deep Learning book and course for his invaluable guidance in applying deep learning with python.

References

[1] H. Anderson, A. Treptow, and T. Duckett, "Self-localization in non-stationary environments using omni-directional vision," *Robot. Auton. Syst.*, vol. 55, no. 7, pp. 541–551, 2007.

[2] G. López-Nicolás, J. J. Guerrero, and C. Sagüés, "Multiple homographies with omnidirectional vision for robot homing," *Robot. Auto. Syst.*, vol. 58, no. 6, pp. 773–783, Jun. 2010.

[3] W. L. D. Lui and R. Jarvis, "Eye-full tower: A GPU-based variable multibaseline omnidirectional stereovision system with automatic baseline selection for outdoor mobile robot navigation," *Robot. Auto. Syst.*, vol. 58, no. 6, pp. 747–761, Jun. 2010.

[4] J. Lim and N. Barnes, "Estimation of the epipole using optical flow at antipodal points," *Comput. Vis. Image Understand.*, vol. 114, no. 2, pp. 245–253, Feb. 2010.

[5] D. L. Stone, G. Shah, and Y. Motai, "Direct spherical calibration of omnidirectional far infrared camera system," *IEEE Sensors J.*, vol. 19, no. 13, pp. 5285–5298, Jul. 2019.

[6] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics," *Int. J. Robot. Res.*, vol. 28, no. 2, pp. 149–171, Feb. 2009.

[7] S. Aksoy, "Spatial techniques for image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2097–2111, Jan. 2008.

[8] O. Helwich and C. Wiedemann, "Object extraction from high-resolution multisensor image data," in *Proc. 3rd Int. Conf. Fusion Earth Data*, 2000, pp. 1–11.

[9] D. Landgrebe, *Information Extraction Principles and Methods for Multispectral and Hyperspectral Image Data*. West Lafayette, IN, USA: Purdue Univ., 1998.

[10] M. Bunkei, Y. Y. C. Wei Jin, O. Yuichi, and Q. Guoyu, "Sensitivity of the enhanced vegetation index (EVI) and normalized difference vegetation index (NDVI) to topographic effects: A case study in high-density cypress forest," *Sensors*, vol. 7, no. 11, pp. 2636–2651, 2007.

[11] M. Bradley, S. M. Thayer, A. Stentz, and P. Rander, "Vegetation detection for mobile robot navigation," Robot. Inst. Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-04-12, 2004.

[12] D. Bradley, R. Unnikrishnan, and J. Bagnell, "Vegetation detection for driving in complex environments," Robotics Inst., Pittsburgh, PA, USA, Paper 52, 2007. [Online]. Available: http://repository.cmu.edu/robotics/52

[13] K. Bhoyar and O. Kakde, "Color image segmentation based on JND color histogram," *Int. J. Image Process. (IJIP)*, vol. 3, no. 6, pp. 283–292, 2010.

[14] A. Briggs, C. Detweiler, P. Mullen, and D. Sharstein, "Scale-space features in 1D omnidirectional images," in *Proc. OMNIVIS*, 2004, pp. 115–126.

[15] A. Pretto, E. Menegatti, Y. Jitsukawa, R. Ueda, and T. Arai, "Image similarity based on discrete wavelet transform for robots with low-computational resources," *Robot. Auto. Syst.*, vol. 58, no. 7, pp. 879–888, Jul. 2010.

[16] M. Taiana, J. Santos, J. Gaspar, J. Nascimento, A. Bernardino, and P. Lima, "Tracking objects with generic calibrated sensors: An algorithm based on color and 3D shape features," *Robot. Auto. Syst.*, vol. 58, no. 6, pp. 784–795, Jun. 2010.

[17] X. Zhang, J. Zhang, C. Li, C. Cheng, L. Jiao, and H. Zhou, "Hybrid unmixing based on adaptive region segmentation for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 7, pp. 3861–3875, Jul. 2018.

[18] A. A. L. M. Rankin Huertas Matthies Bajracharya, C. Assad, and S. P. G. Brennan Bellutta Sherwin, "Unmanned ground vehicle perception using thermal infrared cameras," *Proc. SPIE*, vol. 8045, May 2011, Art. no. 804503.

[19] D. Wolf and G. Sukhatme, "Activity-based semantic mapping of an urban environment," *Robotic Embedded Syst., Auton. Robots*, vol. 19, no. 1, pp. 53–65, 2004.

[20] A. K. Mishra and Y. Aloimonos, "Visual segmentation of 'Simple' objects for robots," in *Robotics: Science and Systems*. Los Angeles, CA, USA, Jun. 2011, pp. 30–38.

[21] J. Cruz Ortiz, "Visual servoing for an omnidirectional mobile robot using the neural network-Multilayer perceptron," in *Proc. Workshop Eng. Appl.*, May 2012, pp. 1–6.

[22] M. Figueiredo, S. Botelho, P. Drews, and C. Haffele, "Spatial and perceptive mapping using semantically self-organizing maps applied to mobile robots," in *Proc. Brazilian Robot. Symp. Latin Amer. Robot. Symp.*, Oct. 2012, pp. 245–250.

[23] K.-T. Song and C. C. Chang, "Reactive navigation in dynamic environment using a multisensor predictor," *IEEE Trans. Syst., Man Cybern., B (Cybern.)*, vol. 29, no. 6, pp. 870–880, 1999.

[24] D. Kim, J. Sun, S. Min Oh, J. M. Rehg, and A. F. Bobick, "Traversability classification using unsupervised on-line visual learning for outdoor robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 518–525.

[25] C. C. Chang and K.-T. Song, "Environment prediction for a mobile robot in a dynamic environment," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 862–872, Dec. 1997.

[26] A. Coates, A. Karpathy, and A. Ng, "Emergence of object-selective features in unsupervised feature learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 2681–2689.

[27] A. Coates, H. Lee, and A. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTAT)*, 2011, pp. 215–223.

[28] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[29] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: Association for Computing Machinery, 2008, pp. 1096–1103, doi: 10.1145/1390156.1390294.

[30] H. Lee, R. Battle, A. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Proc. NIPS*, 2006, pp. 801–808.

[31] S. Hao, W. Wang, Y. Ye, T. Nie, and L. Bruzzone, "Two-stream deep architecture for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2349–2361, Apr. 2018.

[32] W. Song, S. Li, L. Fang, and T. Lu, "Hyperspectral image classification with deep feature fusion network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3173–3184, Jun. 2018.

[33] I. Halatci, C. A. Brooks, and K. Iagnemma, "Terrain classification and classifier fusion for planetary exploration rovers," in *Proc. IEEE Aerosp. Conf.*, Mar. 2007, pp. 1–11.

[34] X. Zhuo, F. Fraundorfer, F. Kurz, and P. Reinartz, "Building detection and segmentation using a CNN with automatically generated training data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Valencia, Spain, 2018, pp. 3461–3464, doi: 10.1109/IGARSS.2018.8518521.

[35] C. Craye, D. Filliat, and J.-F. Goudou, "Exploration strategies for incremental learning of object-based visual saliency," in *Proc. Joint IEEE Int. Conf. Develop. Learn. Epigenetic Robot. (ICDL-EpiRob)*, Aug. 2015, pp. 13–18.

[36] C. Siagian and L. Itti, "Biologically inspired mobile robot vision localization," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 861–873, Aug. 2009.

[37] A. Kelly *et al.*, "Toward reliable off road autonomous vehicles operating in challenging environments," *Int. J. Robot. Res.*, vol. 25, nos. 5–6, pp. 449–483, May 2006.

[38] F. Gao, Y. Xun, J. Wu, G. Bao, and Y. Tan, "Navigation line detection based on robotic vision in natural vegetation-embraced environment," in *Proc. 3rd Int. Congr. Image Signal Process.*, Oct. 2010, pp. 2596–2600.

[39] M. Veres, G. Lacey, and G. W. Taylor, "Deep learning architectures for soil property prediction," in *Proc. 12th Conf. Comput. Robot Vis.*, Jun. 2015, pp. 8–15.

[40] J. Nagi *et al.*, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Nov. 2011, pp. 342–347.

[41] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[42] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016.

[43] A. Rosebrock, *Deep Learning for Computer Vision With Python*, 1st ed. Baltimore, MD, USA: PyImageSearch, 2017.

[44] F. Chollet, *Deep Learning With Python*. Shelter Island, NY, USA: Manning Publishing Co., 2018.

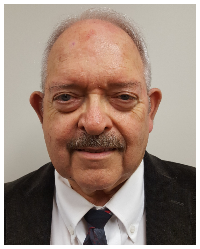
[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*. [Online]. Available: http://arxiv.org/abs/1512.03385

[47] D. L. Stone, G. Shah, Y. Motai, and A. J. Aved, "Vegetation segmentation for sensor fusion of omnidirectional far-infrared and visual stream," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 2, pp. 614–626, Feb. 2019.

[48] E. Raczko and B. Zagajewski, "Comparison of support vector machine, random forest and neural network classifiers for tree species classification on airborne hyperspectral APEX images," *Eur. J. Remote Sens.*, vol. 50, no. 1, pp. 144–154, Jan. 2017, doi: 10.1080/22797254.2017.1299557.

[49] G. Omer, O. Mutanga, E. M. Abdel-Rahman, and E. Adam, "Performance of support vector machines and artificial neural network for mapping endangered tree species using WorldView-2 data in Dukuduku Forest, South Africa," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 4825–4840, Oct. 2015.



David L. Stone (Member, IEEE) received the Bachelor of Science degree in electrical engineering with an emphasis on computer architecture from Purdue University, West Lafayette, IN, USA, in 1973, the master's degree in mechanical engineering and the Ocean Engineer degree in naval architecture from the Massachusetts Institute of Technology, Cambridge, MA, USA, both in 1982, and the Ph.D. degree in electrical and computer engineering for robotic perception from Virginia Commonwealth University (VCU), Richmond, VA, USA, in 2019.

He spent 27 years in the U.S. Navy transitioning from an Electronics Technician 1st Class to an Officer in the Engineering Duty Corps. While in the Navy, he worked on numerous engineering projects including the Navy's Large Scale Vehicle autonomous submarine for testing of the SEAWOLF and VIRGINIA class submarines. After retiring in 1994, he worked as a Senior Engineer and the Program Manager with SAIC, Bayview, ID, USA, for the Electronics and Data Analysis Support Contract for NSWC CD and the LSV Program. Starting his own business, he served as the President and the Chief Engineer with the Unmanned Systems Technology Lab, Inc., Coeur d' Alene, ID, USA, doing work in both underwater and ground robotics for DARPA and ONR. He worked as a Lead Systems Engineer for the Army Command Post Platform Project with SprayCool, Spokane, WA, USA. He worked for Unmanned and Robotic Systems Branch, H42, NSWC, Dahlgren, VA, in May 2009, and is supporting the Marine Corps Warfighting Lab Technology Division Ground Combat Element as a Robotics SME.

Sumved Ravi received the B.Sc. degree in electrical engineering with a concentration in control, robotics and autonomy, as well as computer engineering with a concentration in machine learning from Virginia Tech, Blacksburg, VA, USA, in 2020, where he is pursuing the M.Sc. degree in computer engineering with a focus on deep learning and computer vision.

His research interests include object characterization, tracking, video captioning, network compression, and artificial general intelligence.

Emrah Benli (Member, IEEE) received the B.Sc. degree in electronics and communication engineering from Kocaeli University, Kocaeli, Turkey, in 2009, the M.Sc. degree in electrical engineering from Clemson University, Clemson, SC, USA, in 2013, and the Ph.D. degree in electrical and computer engineering from Virginia Commonwealth University, Richmond, VA, USA, in 2017.

He was a Post-Doctoral Fellow specializing in autonomous mobile robotics with the U.S. Army Research Laboratory's Computational and Information Sciences Directorate, Adelphi, MD, USA, in 2018. He is an Assistant Professor of Electrical and Electronics Engineering with Karadeniz Technical University, Trabzon, Turkey. His research interests include intelligent systems, computer vision, artificial intelligence, multimodal sensory, robotic system design and control, and human–robot interaction.

Yuichi Motai (Senior Member, IEEE) received the B.Eng. degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991, the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2002.

He is an Associate Professor of Electrical and Computer Engineering with Virginia Commonwealth University, Richmond, VA, USA. His research interests include the broad area of sensory intelligence; particularly in medical imaging, pattern recognition, computer vision, and sensory-based robotics.