Approved for public release: distribution unlimited.

Metadata Enabled Contextual Sensor Fusion for Unmanned Aerial System-Based Explosive Hazard Detection

Matthew Deardorff[a], Brendan Alvey[a], Derek T. Anderson[a], James M. Keller[a], Grant Scott[a], Dominic Ho[a], Andrew Buck[a], Clare Yang[b], and Brad Libbey[b]

[a]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia MO, USA
[b]U.S. Army DEVCOM C5ISR Center, Fort Belvoir, VA, USA

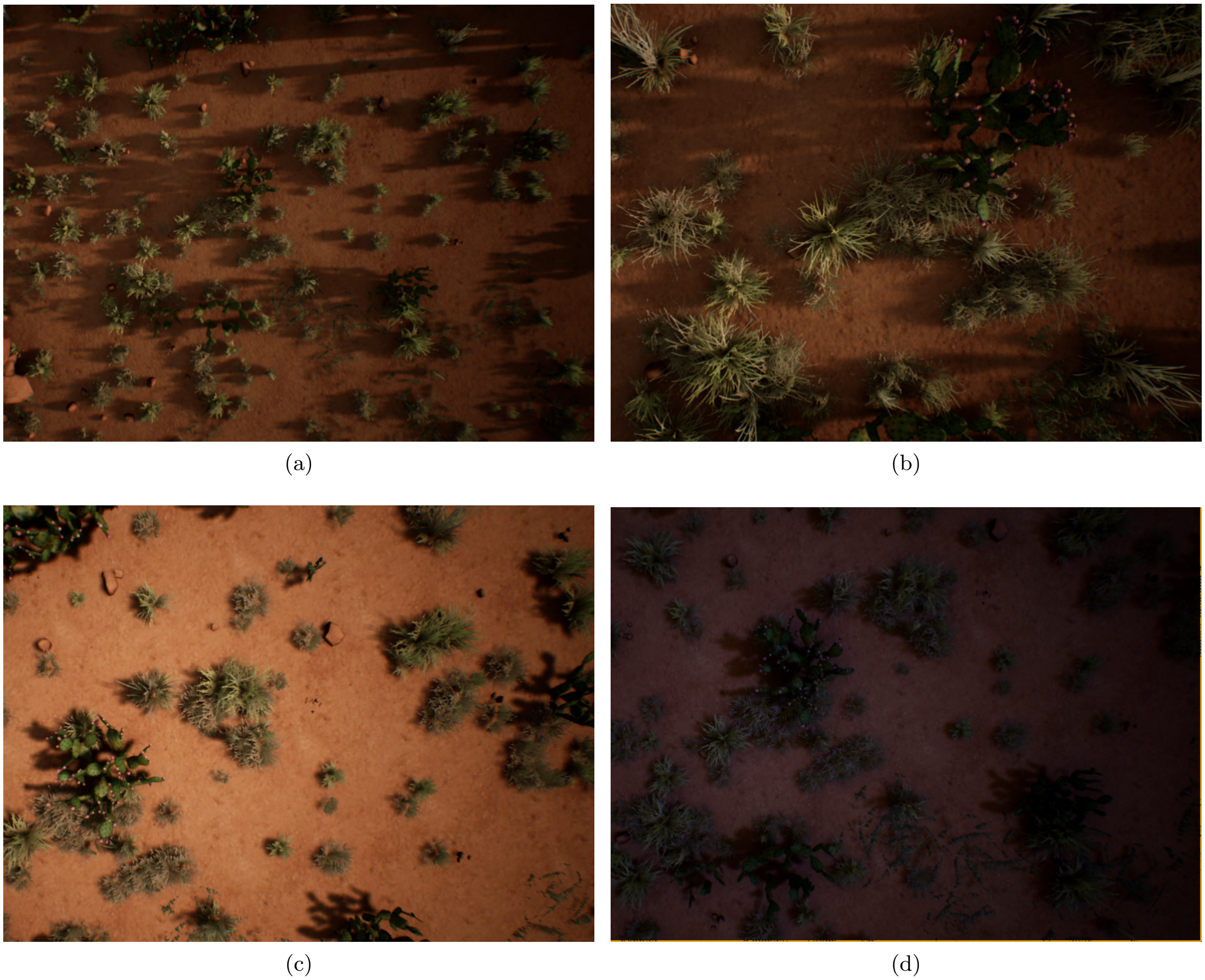
ABSTRACT

Numerous real-world applications require the intelligent combining of disparate information streams from sensors to create a more complete and enhanced observation in support of underlying tasks like classification, regression, or decision making. An often overlooked and underappreciated part of fusion is context. Herein, we focus on two contextual fusion challenges, incomplete (limited knowledge) models and metadata. Examples of metadata available to unmanned aerial systems (UAS) include time of day, platform/sensor position, etc., all of which have a potentially drastic impact on sensor measurements and subsequently our decisions derived from them. Additionally, incomplete models limit machine learning, specifically under-sampling of training data. To address these challenges, we investigate contextually adaptive online Choquet integration. First, we cluster and partition the training metadata. Second, a single machine learning model is trained per partition. Third, a Choquet integral is learned for the combination of these models per partition. Fourth, at test/run time we compute the degree of typicality of a new sample to our known contexts. Fifth, our trained integrals are decomposed into a bag of underlying aggregation operators and a new contextually relevant operator is imputed using a combination of the metadata clustering and observation statistics of the integral variables. This process enables machine learning model selection, ensemble fusion, and metadata outlier detection, with subsequent mitigation strategy identification or decision suppression. The above ideas are demonstrated on explosive hazard detection using surrogate data simulated by the Unreal Engine. In particular, the Unreal Engine is used because it provides us with flexibility to explore the proposed ideas across a range of diverse and controlled experiments. Our preliminary results show improved performance for fusion in different contexts and a sensitivity analysis is performed with respect to metadata degradation.

1. INTRODUCTION

The task of detecting and classifying explosive hazards (EH) from unmanned aerial systems (UAS) is a difficult one, in part due to the drastically varied environments and platform conditions one can expect to operate in/across. Detection in a hot desert at noon is a significantly different problem than detection in a frozen tundra at night. Detection from a UAS with a nadir sensing angle at 10 feet is different from a UAS with a sensor slant angle at 100 feet. Furthermore, the sensors used on UASs–e.g., RGB, IR, LiDAR, multi-spectral, etc.–all experience different sensor phenomenology depending on the environmental conditions and material properties of sensed objects. Figure 1 is an example that highlights environment and UAS (platform) variation. Herein, we refer to the above variations as *contexts*, as they are sources of information which we can use to enhance the performance of an underlying task like EH detection. In this paper, we propose an online and adaptive ensemble-based fusion scheme for EH detection that is driven by environment and platform metadata.

Before we delve into UAS-based EH detection (EHD), we briefly discuss related efforts. EHD technologies vary drastically. An early and well-known technique is the so-called "metal detector", which can be used to detect metal in the ground. However, one limitation with this form of detection is that explosive threats that contain low amounts of metal may go undetected. Increasing the sensitivity of the device does not necessarily



(a)

(b)

(c)

(d)

Figure 1: Detection and localization algorithms are tasked with understanding objects in a variety of *contexts*, requiring robustness across factors like scale, color, illumination, and texture. Often, even a single location can look very different depending on platform altitude, look angle, time of day, etc. However, this information often goes unused in algorithms. The proposed contextual fusion scheme attempts to determine proper strategies based on metadata features which help inform context.

counteract this, as the number of false alarms would likely dramatically increase. To increase the robustness of detection, many different combinations of sensing methods have and are being explored, such as infrared (IR), ground penetrating radar (GPR), electromagnetic induction (EMI), and hyperspectral imaging (HSI), to name a few. The two predominant approaches to date for detecting explosives is vehicle-mounted detectors and hand-held detectors. While the latter is predominantly used in a downward looking fashion, the prior comes in a multitude of forms, e.g., forward looking,[1] downward looking,[2] and even side looking.[3] Herein, we focus on a UAS platform for EHD. Advantages of UAS, versus hand-held or ground vehicle deployment is it keeps humans at safer standoff distances and a UAS can in theory act like each of the above technologies. That is, it has the potential to search wide areas and dynamically interrogate regions of interest, likely through the use of a squad or swarm of UASs, with different sensors at different look angles. In this article, we limit our analysis to the use of a single UAS with multiple imaging and position sensors.

Adaptive fusion is not a new idea. For example, in Ref. 4, Hichem, Gader, et al. proposed a creative algorithm, context extraction for local fusion (CELF). We highlight and discuss this algorithm because we also use the Choquet integral (ChI). In particular, Hichem combined the fuzzy C-means (FCM) clustering algorithm and the ChI. They formulated a single joint optimization. Hichem partitions the input space based on the features to be fused. Our work, aka the current article, differs as we initialize contexts based on otherwise unused metadata features such as altitude and temperature and learn independent operators from subsets of data. We also approximate the entire integral, aka all the underlying capacity variables, of which there are 2^N for N inputs. Hichem instead focuses on the "densities", i.e., the capacity defined on only the singletons, and an imputation strategy (the Sugeno λ fuzzy measure). We focus on learning the entire capacity because the tuples beyond the singletons capture interaction between sources. This is something we expect to occur and it can result in performance gain. Last, Hichem's fusion is driven by clustering. Herein, we exploit our recent integral transfer learning[5,6] and data-driven eXplainable AI (XAI) methods.[7,8] XAI allows us to identify what parts of a model (integral) were not approximated (sufficiently) from training data. Integral transfer learning allows us to transfer fusions, or parts of fusion, across integrals. In contrast to prior methods, our proposed method provides a method to measure similarity of new samples relative to prior contexts and determine the sufficiency of the fusion operator being imputed. These similarity methods are important as our goal is to optimize the aggregation operator based on information previously observed.

We also explore the use of the Unreal Engine[9] to create synthetic imagery. The graphical fidelity offered by these simulated environments proves as a useful surrogate for the otherwise difficult task of assembling large amounts of varied, UAS-captured data. Advantages include training real models from simulated data and rapid prototyping and experimenting with ideas that can later be transfered to real world experiments and solutions.

1.1 Machine Learning Models Derived from Limited Data Sample Sets

Data is king in modern machine learning. The performance of neural networks and other supervised learning models are intimately linked with the kind, quality, and diversity of training data provided. In a perfect world we could assume that good quality data can be obtained with enough time and patience, but this is rarely the case. It is in our interest to develop well performing classification models that have been exposed to only a limited amount of training data.[6,10] This is especially relevant in the domain of UAS based vision as it can be difficult to obtain large amounts of appropriate aerial data.

The problem of limited training data is one which informs how the rest of this architecture is structured, and must be considered at every step. One known problem caused by small amounts of training data is *overfitting*. Overfitting occurs when a learning model memorises the solutions to training data but is unable to generalize to data it has not seen before. This is in part due to the training data lacking adequate diversity, as the training data does not represent all possible variations of data that might be discovered. Our current article attempts to mitigate the overfitting problem by relying on a collection of niche experts, as opposed to a single model which can universally solve the problem. As a result, models are only expected to perform well on data similar to what has been seen before and their use is restricted when performance expectations are low.

Another problem encountered due to limited training data is specifically tied to our fusion operator of choice, the ChI. As described later, the ChI partitions the input space based on the sorting of the inputs, where each unique sort results in a different method of combination. Because of this, we ideally would observe every possible sort in the training data so that an optimization algorithm is able to estimate all the values that are required. This is often not the case for a number of reasons. First, it can simply be difficult to encounter each sort through random chance. For N inputs there exist $N!$ possible sorts, meaning an adequately sized training set is required just to get one sample from each sort. A second reason the observed sorts are important is specifically tied to the domain of fusing *strong learners*. A strong learner is a learning model which usually produces only extreme (strong) values. For example, a strong classifier would only label detections as 0 (not the class) or 1 (is the class), but would rarely label something as 0.5 to denote uncertainty. Due to this, it is often the case that all models to be fused agree on a class label of either 0 or 1. Thus, the only sort encountered is the default from when all values are the same. This heavily biases the ChI training procedure, as it is possible that nearly all observed data consists of only a few unique walks. If unencountered sorts ever show up later in testing data, the operators will be poorly optimized to handle them. Our method attempts to mitigate this problem by transferring learned values from an integral that has observed a particular sort.

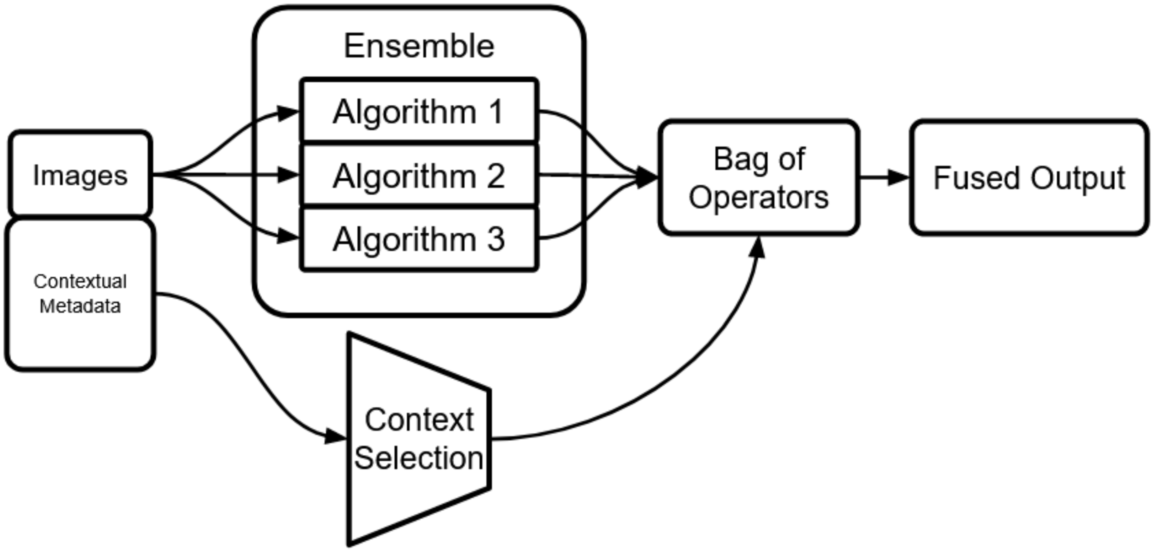


Figure 2: The general flow of images and metadata in our ensemble. Multiple algorithms are treated as sources of evidence to be fused together, while metadata such as altitude, temperature, and time of day inform the system how to construct the best possible aggregation operator.

1.2 Ensemble of Neural Networks

A common technique to mitigate the reliance on a single black-box neural network is to train multiple networks which operate in parallel on the data. While this goes by different names in the community, we refer to it herein as an ensemble neural network. Each of the networks produces its own estimate of the target value, before each of the estimates are aggregated back into a single score. The precise method of aggregation depends on the architecture, though our method uses the ChI, in part due to it's capability of producing human-readable explanations of the fusion. This article creates an ensemble of networks with homogeneous architectures trained on varying subsets of data. A different popular technique in ensemble architectures is to vary the architecture of the individual networks (depth, number of parameters, etc.) but provide each network with complete data. The reader can refer to Refs. 11–14 for our recent publications on ensembles of heterogeneous architecture neural networks for broad area scanning, land classification, and object detection in remote sensing. Figure 2 illustrates the flow of data and metadata in the ensemble architecture proposed herein. The pieces of this ensemble are described in greater detail in following sections.

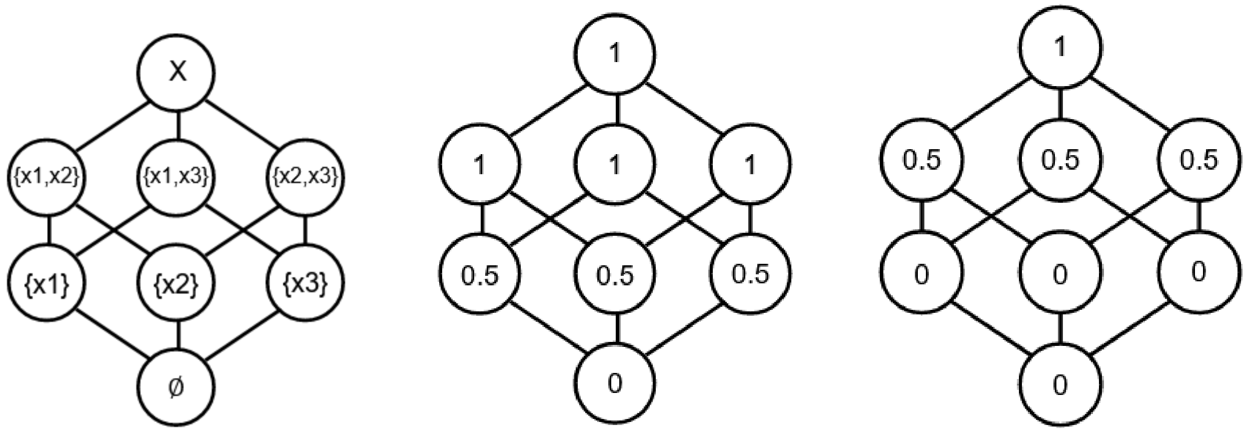
2. METHODS

2.1 Fuzzy Measure and Fuzzy Integral

The fuzzy integral (FI) is a well studied tool in information fusion which defines a family of nonlinear operators. The integral is evaluated on a fuzzy measure (FM), $g : 2^X \to R^+$, which is a function that has two properties on finite X: (i) (boundary condition) $g(\emptyset) = 0$, and (ii) (monotonicity) if $A, B \subseteq X$, and $A \subseteq B$, then $g(A) \leq g(B)$. The Choquet integral (ChI) is a *type* of FI,[15] given by

$$\int \mathbf{h} \circ g = C_g(\mathbf{h}) = \sum_{j=1}^{N} h_{\pi(j)}(g(A_{\pi(j)}) - g(A_{\pi(j-1)})), \tag{1}$$

where $\mathbf{h}$ is the integrand ($h(\{x_i\}) = h_i$ is the input from source i), $A_{\pi(j)} = \{x_{\pi(1)}, \ldots, x_{\pi(j)}\}$, $g(A_{\pi(0)}) = 0$, and π is a sort such that $h_{\pi(1)} \geq h_{\pi(2)} \geq ... \geq h_{\pi(N)}$. In our case, $h_{\pi(j)}$ is the jth largest return out of all algorithms, and $a \subseteq A$, $g(a)$ denotes the "worth" of a subset of algorithms. Thus, the ChI fuses evidence from each source based on the worth of a subset of sources.



(a) Set-Labeled Hasse (b) Optimistic Fusion (c) Pessimistic Fusion

Figure 3: Hasse diagrams depicting different strategies of fusion for $N = 3$ inputs. An optimistic fusion like the one depicted in 3b averages the two largest input values. A pessimistic operator 3c averages the two smallest values. In algorithm fusion it is common to see pessimistic operators due to their redundancy as all algorithms must agree on a high value, i.e., unanimous consent.

It is relevant to note that the ChI is an operator which can be learned from data using various solvers. For example, in Ref. 16 we use quadratic programming (QP), in Ref. 11 we proposed constraint free full FM gradient descent optimization for supervised neural networks, and in Ref. 17 we proposed an evolutionary algorithm for efficient genetic operators on non-convex optimization surfaces. In this paper, we use the QP to learn a number of ChIs, trained on subsets of the data, which can be selected from based on what we believe the context to be.

A convenient way to visualize the ChI is in the form of its underlying Hasse diagram, where nodes in the diagram represent the g values of the power set of A in lexicographic order from bottom to top, left to right. For example, if $X = \{1, 2, 3\}$ the lexicographic ordering of the power set $P(X)$ is

$$P(X) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$

The edges in the diagram represent monotonicity constraints, meaning nodes in the upper layers are greater than or equal to the nodes connected below them. A *walk* up the Hasse diagram refers to a given sort on $\mathbf{h}$ and the resulting path taken from the bottom of the diagram to the top. Therefore, each walk defines a unique fusion operation the ChI is capable of. Figure 3 depicts example diagrams which are possible fusion strategies for three sources ($N = 3$).

2.2 Context Matters

The discrete ChI described above partitions the input space based on the sorting of inputs and each partition results in a different fusion operator. One way to think of these partitions is that they provide *context* as to what operator is most appropriate. An example interpretation of this for our current paper on UAS-based detection of EHs using multiple neural net algorithms is: "if Algorithm 1 has the greatest return, listen primarily to it. Otherwise, take the average of all the algorithms". We call this kind of context the *internal context*, as it is based solely on the data that is directly being fused. Specifically, Equation 2.1 informs us that each internal operator context is a linear convex sum (LCS) function, when $g(\emptyset) = 0$ and $g(X) = 1$.

However, there is more than just internal context in problems such as ours. Consider the task of EHD from a UAS. There are wildly different conditions in which the UAS might be flown, such as high altitudes, low altitudes, bright days, or dark nights. These are all normal operating conditions for such a system, yet the sensory feedback in each of these conditions will be distinct. As a result, the algorithms that we use on this data must be robust to these variations. This article aims to better handle this kind of context, what we call the *external context*, of our fusion problem. Our method attempts to identify these unique external contexts by clustering the metadata

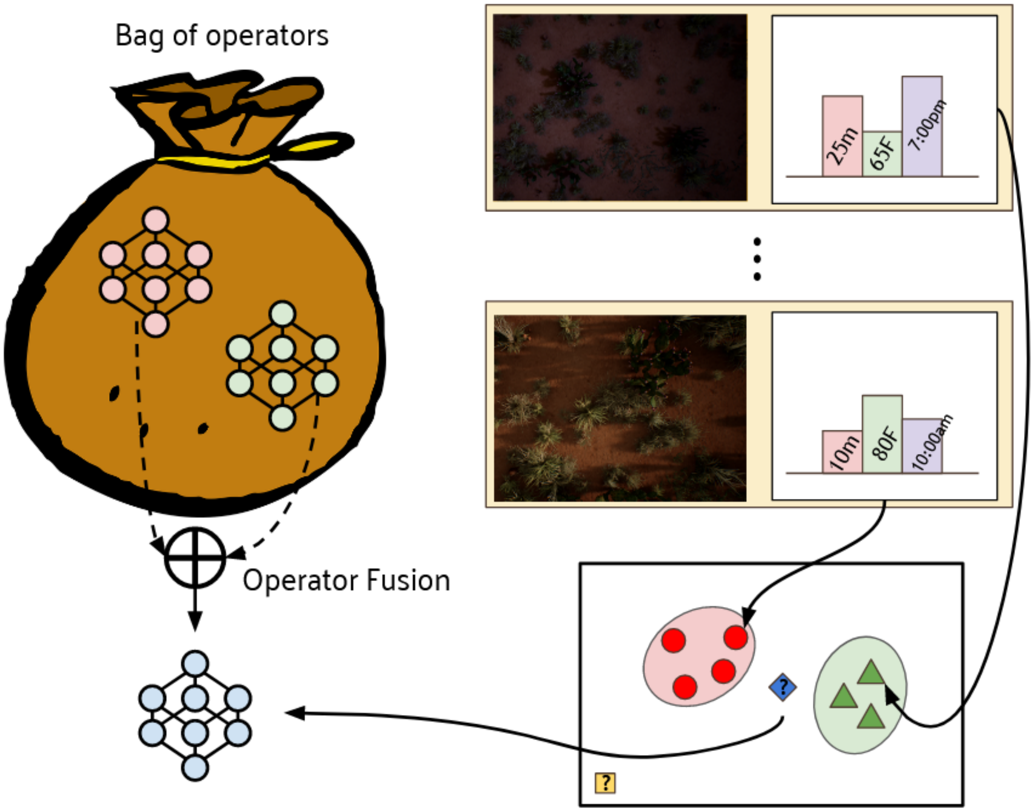


Figure 4: Illustration of the propose methodology. Metadata feature vectors are generated from training data and they are clustered to define initial contexts (red circles and green triangles). In this diagram we show an example image and the prototype per cluster. Next, a different ChI that combines a set of neural network classifiers is built per context (red and green Hasse diagrams). When a new sample (observation) belongs to a known context, the appropriate ChI is used. However, if a new sample, e.g., blue diamond, does not belong to a known context but it is similar to known contexts, then a new operator is built on the fly. In the event that a new sample is extremely different from anything that we have seen before, e.g., the yellow box outlier, then the system can decide to take no action or an operator can be built if the system is expected to always operate.

obtained from the UAS (platform) and environment. Specifically, we use the GPS reported altitudes, recorded temperatures, and time of day as initial features to identify unique external contexts. Figure 4 illustrates our scheme of clustering metadata and using them to train unique fusion operators.

2.3 Metadata Feature Encoding

As the following metadata features will be used to inform the system of which context to associate the data with, it is important to consider their encoding. A common problem resulting from the use of disparate feature types is that one feature can dominate the space, e.g., have notably higher magnitudes. In our case, if we assume that the range of observed temperatures is on average larger than the range of observed altitudes, then the distance between temperatures will predominantly drive the distance measure. While there are other ways to handle this using techniques such as categorical encodings, a simple solution is to normalize the values (denoted as z below) on a scale of [0,1] based on minimum and maximum observed values,

$$z_{scaled} = \frac{z - z_{min}}{z_{max} - z_{min}}. \tag{2}$$

Special attention should also be paid to how the time of day is encoded, as it is a cyclical feature. Consider what happens if time of day was encoded as a scalar value in the range 0 to 23 hours (12am to 11pm). If we measure the distance from $t = 23$ to $t = 1$ ($11 : 00$pm to $1 : 00$am), the Euclidean distance is 22, though clearly those two time periods are only two hours apart. A simple yet clever way to avoid this problem is to split the time feature into two values given by

$$t_{sin} = \sin\left(\frac{2\pi t}{23}\right), t_{cos} = \cos\left(\frac{2\pi t}{23}\right). \tag{3}$$

When these two values are plotted as (x, y) pairs in the range $[0, 23]$, the result is a circle. This makes it a more appropriate encoding for use with Euclidean distance, as it now mimics distance on an actual clock, i.e., $t = 0$ and $t = 23$ are adjacent, while any two values offset by 12 hours maximize the distance. Note, in this paper we explore a few metadata. In future work we will investigate the inclusion of more metadata and their respective pleasing semantic conditioning.

2.4 Determining Initial Contexts Through Clustering

As already discussed, our ensemble of neural networks is driven by context. To this end, we cluster the training metadata features into an initial set of contexts via the possibilistic c-means (PCM) algorithm.[18] The PCM is a mode seeking method that operates on a finite set of M samples $Z = \{\mathbf{z}_1, ..., \mathbf{z}_M\}$ relative to a specified number of c clusters. Unlike the k-means clustering algorithm, which is a crisp partitioning technique (i.e., every sample belongs to one, and one only, cluster), the PCM is a mode seeking algorithm. The PCM returns c clusters, which depending on the choice of underlying metric (e.g., Euclidean, Mahalanobis distance, GK metric, etc.) results in c prototypes, e.g., $C = \{\mathbf{c}_1, ..., \mathbf{c}_c\}$, and a partition matrix $[U]_{ik} = u_{ik}, i = 1, ..., C, k = 1, ..., M$, where u_{ik} is the typicality of sample $\mathbf{z}_k$ to cluster i. Unlike the k-means algorithm, the PCM allows samples to belong fully to multiple clusters and outliers can now be represented and detected. The PCM typicality degrees are especially useful at evaluation time, as it gives us a degree to which we believe a new data point belongs to a known context (cluster). As described later, we adapt our fusion strategy for new data (UAS observations) based on how similar it is to what has been seen before. In our implementation of the PCM, we use Euclidean distance and we initialize the cluster centers with the fuzzy c-means algorithm output because it helps us estimate PCM bandwidth parameters and it provides robustness over random initialization.

A problem ever present in all clustering algorithms is determining an optimal value for c. Herein, we use the fuzzy partition coefficient[19] (PC), an internal cluster validity index. The PC attempts to measure how well a set of data was partitioned based on the membership values of each class given by

$$F_c(U) = \frac{tr(U * U^T)}{M}, \tag{4}$$

where U is the fuzzy partition matrix segregated into c classes, $*$ is matrix multiplications and $tr()$ is the trace, or sum of squared diagonals. A desired c can be selected from an index like the PC by looking for the maximum (or minimum) index value, or a trend (e.g., elbow) in the c plot. While the PC is used herein, it should be noted that there are more sophisticated internal (Xie and Beni index, Dunn, DBI, etc.) and external (Rand, etc.) cluster validity measures in the community, e.g., see Ref. 20. If the reader desires to implement and use the methodologies contained herein, we recommend that a more robust cluster validity index be used.

2.5 Realtime Fusion

The above sections describe a set of offline computations on training data. The result is a set of contexts, neural classifiers (one per context), and subsequent aggregation operators (one ChI per context). This section outlines an online (aka runtime) selection mechanism to determine what contexts a new sample belongs to based on the typicality values provided by the PCM algorithm.

The selection process we developed breaks down into three distinct cases. The first case is when the data to be evaluated is highly typical of one and only one existing context, meaning we believe we have an appropriate fusion operator to use. The second case occurs when the data to be evaluated is highly atypical compared to all known contexts, meaning we are operating in an unknown context and will subsequently resort to using a default fusion operator. Herein, we explore the idea of a system taking an action, but a user could instead take no action because we are unable to predict how the system will respond. The third and final case occurs when the data to be evaluated belongs to more than one context. This is a case where we will fuse multiple operators together to create a more appropriate adaptive fusion scheme.

The selection process above can be defined for data point z_i, where u_{ij} is the typicality of z_i in cluster j, and α and β are user defined upper and lower typicality thresholds respectively. The selection function is

$$\mathbf{g} = \begin{cases} g_k & u_{ik} \geq \alpha \text{ and } \forall j, j \neq k, u_{ij} < \alpha \\ g_{\text{default}} & \forall j, u_{ij} < \beta \\ combine(\mathbf{u}, g_1, ..., g_N) & \text{else,} \end{cases}$$

where condition one says pick a single FM/ChI when we are in a known context. Condition two is how we respond to a metadata outlier and condition three outlines the fusion of our fusions from metadata. The above scheme still leaves a few questions. First, what operator do we choose in case two? This is the case where the system is exposed to what appears to be a thus-far unseen context. We explore multiple methods, including a simple mean average of the network confidences and averaging the operators from all previously observed contexts. The simple mean is a natural place to start, as it credits equal worth to each of the sources to be fused. This is useful as it makes no assumptions about the worth of individual sources in unseen contexts, though this is also the method's weakness. If there is a clear pattern in the fusion strategies consistent across all contexts then the simple average will disregard this, throwing away information that could be useful as a default fusion strategy. The second method explored attempts to handle this problem by averaging the set of trained ChI operators. Here we define the average of a set of operators to mean calculating the average value on a per-node basis in the Hasse diagram. This produces a fusion scheme which retains any dominant fusion strategies common across contexts, while maintaining the monotonicity constraints required by the ChI. If there is no obvious fusion scheme across all contexts (such as being generally optimistic or favoring a particular source), this averaged operator produces an operator that is in a way smoothed, and pulled closer to an operator that resembles the mean[*].

This solution inspires the combination method we use for case three (see Figure 5). In this case, our clustering is tight enough that a new data point can reasonably be considered to be in one of multiple contexts. To resolve the ambiguity, the operators in question are combined through a weighted average where the weight is determined by the relative strength of the typicality values. The $combine(\mathbf{u}, g_1, ..., g_N)$ function above is

$$\mathbf{g} = \sum_{i=1}^{N} \frac{u_{ik}}{T(\mathbf{u})} g_i, \tag{5}$$

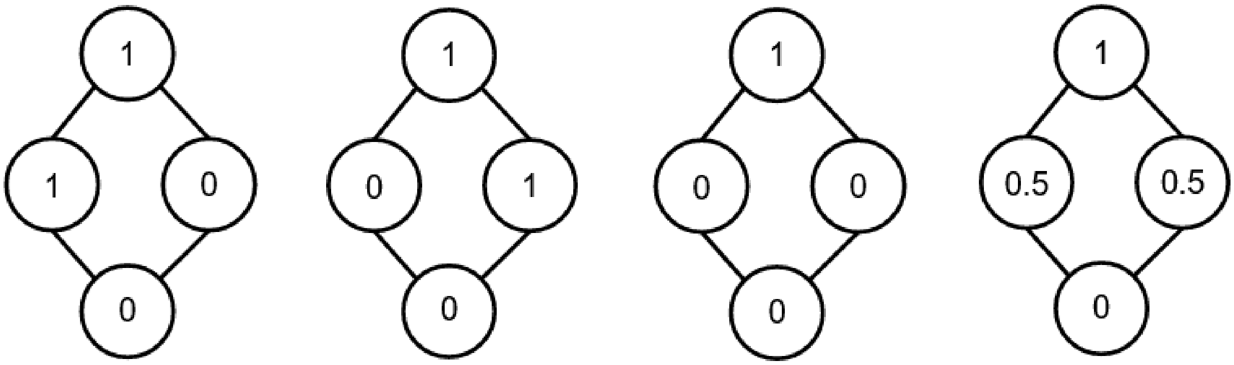
where γg ($\gamma \in [0, 1]$) is defined as $\gamma g(A), \forall A \in 2^X$, $g = g_i + g_j$ is defined as $g(A) = g_i(A) + g_j(A), \forall A \in 2^X$, $T(\mathbf{u})$ is the sum total of typicalities for sample k, $T(\mathbf{u}) = \sum_{i=1}^{N} u_{ik}$. Other possible ways to aggregate FMs include the operators we outlined in Ref. 21 relative to evolutionary optimization, a simple t-norm like the minimum or product on each variable, or a set of t-norms and t-conorms outlined in Ref. 22 by Yager.

On a final note, we wish to comment that this is an initial study. That is, the above method only exploits metadata cluster membership values. This helps us build a new operator on-the-fly based on how similar the sample is to our past contexts. In future work we will also look at the internal context in each ChI (runs in the Hasse diagram) and combine it with our probabilistic estimate of how well that operator was supported.[7,8] The idea being, there is no point relying on an operator that has not been sufficiently learned from data. Instead, a method like our ChI transfer learning[5] or similar should be engaged to derive an appropriate data informed internal operator. Last, these two disparate concepts need be combined.

3. PRELIMINARY EXPERIMENTS AND ANALYSIS

In this section we explore our proposed methods on a set of synthetic imagery meant to imitate changes in sensor phenomenology we might expect from use in different UAS environments. Imagery was generated using the Unreal Engine, as it allows automatic data-labeling and complete control of environment parameters. This provides us needed flexibility to explore ideas like adaptive fusion. That is, our methods are not bottle necked by

[*] Assuming uniformly random FM g values



(a) First operator (b) Second operator (c) Minimum combine (d) Average combine

Figure 5: What is a reasonable scheme to combine FMs? 5a and 5b signify fusion schemes which listen entirely to a single source, a result that is likely to happen in our system if a given algorithm performs especially well in a certain context. If we combine based on a minimum operator or allow the quadratic solver to recompute on all data, the result is 5c. This operator is very pessimistic and will require both algorithms to agree on an answer, something that may be unlikely to happen. 5d is the result of a node-wise average, and maintains a degree of worth for individual algorithms.

real world factors like time and ultimately expense of collecting and labeling EH data. In our experiments we use the You Only Look Once version 5 (YOLOv5) network architecture[23] for EH object detection and localization, as it provides estimates of bounding boxes and confidences to fuse across, with a well documented implementation for easy training. We compare our method against a general model which has been exposed to all training data and is not a part of an ensemble. We examine what happens when the system is exposed to contexts that are not present in the training data, as well as good strategies for combining existing operators when the metadata is ambiguous between multiple existing contexts.

It should be noted that we are intentionally not disclosing which environments, EH targets, and EH emplacement strategies we simulated. The targets and environments were determined in conjunction with our US Army Night Vision and Electronic Sensors Directorate (NVESD) collaborators. The targets are above ground objects (versus buried), they have moderate-to-low clutter (e.g., are often partially obscured by natural objects like a bush), we use a generic (aka similar to what you would find on the commercial market) RGB camera, and we believe that objects have enough pixels on target for detection. The goal was not to push the system to extreme breaking points, i.e., camera spectra being insufficient to detect an object or too few of pixels to even have an object that can be detected and discriminated from clutter. The point is to create a challenging and real world achievable problem that we can push to the point of failure and to compare the different avenues outlined herein. Furthermore, the UAS platform conditions were nadir (looking straight down) and a few arbitrary altitude variations were explored. In this article we do not consider all common scenarios, e.g., varying factors like aircraft speed and subsequent motion blur that would result. The point is, exact altitudes, environments, camera parameters, and etc. are just a surrogate herein to test the proposed algorithms. These experiments and environments are not real, but they are set up to mimic similar conditions to data that we have seen to date; meaning they are not overly simple and unrealistic. This paper is not a documentation of YOLOv5 for EHD on specific use cases. We would not report that information due to the real-world EH threat. In summary, what the reader can take away from the following experiments is the relative performance of the algorithms, their variations, and sensitivities.

3.1 Metadata Enabled Fusion versus Single Model

We start by evaluating the proposed algorithms on six sets of synthetic training data, where each dataset represents a different context that a UAS could experience during normal operation. As mentioned above, our goal is generality and diverse training data, not specific operational experiments. Specifically, the training runs

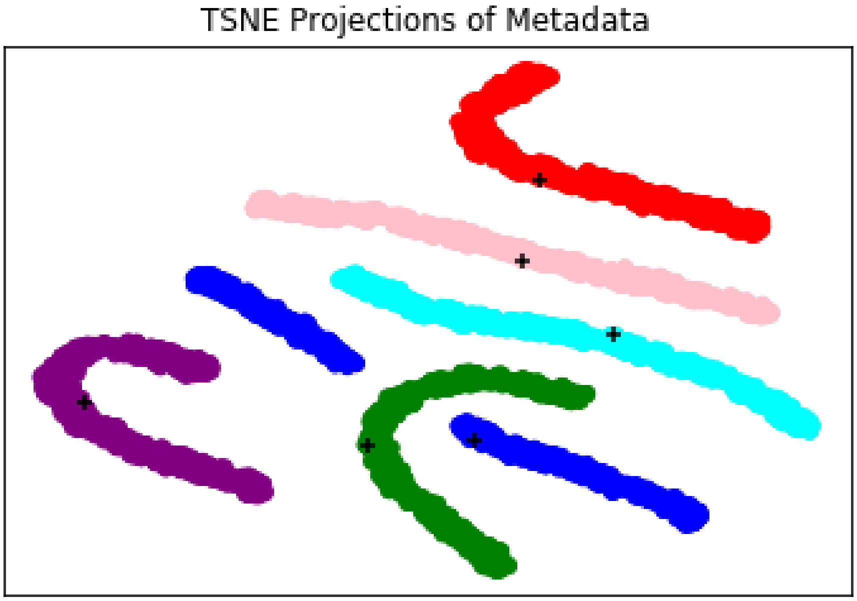


Figure 6: The metadata of our training sets reduced from four to two dimensions by TSNE. Color-coding is provided by PCM assigned clusters. Cluster centers are marked with a plus.

consist of "high" and "low" altitude variants at solar noon (aka no shadows and ideal radiance conditions), afternoon (long shadows, darker), and night (very dark, most difficult) data. Below, these training datasets are referred to as "data set 1", "data set 2", and etc., and test datasets are simply referred to as "test 1" (Test1), "test 2" (Test2), etc. The test data sets contain otherwise unseen data (i.e., not resubstituion) with targets that are under heavy occlusion, shadows, and extreme angles so that they should be sufficiently difficult tests to evaluate our method. We feel like there is no loss of generality in our paper, as one can see performance in context, out of context, and with respect to outlier observations. As each training run has associated higher dimensional metadata (four dimensions herein) that we visualize using the dimensionality reduction techniques t-distributed stochastic neighbor embedding[24] (TSNE). Figure 6 shows the metadata from the six training runs, along with PCM assigned clusterings.

While the clusters are clearly separable in our experiments, when transitioning to the real world we do expect the data to be noisier. This can lead to a larger bandwidth parameter in the PCM algorithm, which will ultimately cause typicalities to increase across the board as the algorithm becomes more relaxed in what it considers a cluster. In short, these experiments are less likely produce the third case in the context selection procedure, as it is unlikely for a given point to have high typicalities across multiple clusters. It can also be noted that the reason the clusters manifest as rope-like in the projection is due to the time of day feature increasing linearly through time while the other features are pulled from a normal distribution. Furthermore, we study this separable problem because it mimics the way that many real world collections occur. That is, data is often collected for a short number of consecutive days in a specific geographic area. We would not expect to have data from twenty four hours a day at all geographic locations. Last, it is our strong belief that if the proposed algorithms do not work for the scenarios explored herein, it is unlikely that they will work for the more challenging scenarios. And as stated above, an advantage of the Unreal Engine is we can generate a lot of data, of which all attributes are known. This is rarely the case in the real world as labeling can be sparse and error prone and documentation is never complete nor perfectly accurate (e.g., amount of cloud cover, temperature at each geospatial location, etc.).

Figure 7 shows the relative performance of our ensemble network (solid lines) compared to a single, out-of-the-box YOLOv5 model (dotted lines). Ideally, a good receiver operating characteristic (ROC) curve, where the x-axis is mistakes and y-axis is the positive detection rate, is the "zero FAR, one PD" (which is almost always achievable in real datasets). Most often, people look for "quick rises" (increase in PD with little to no mistakes)

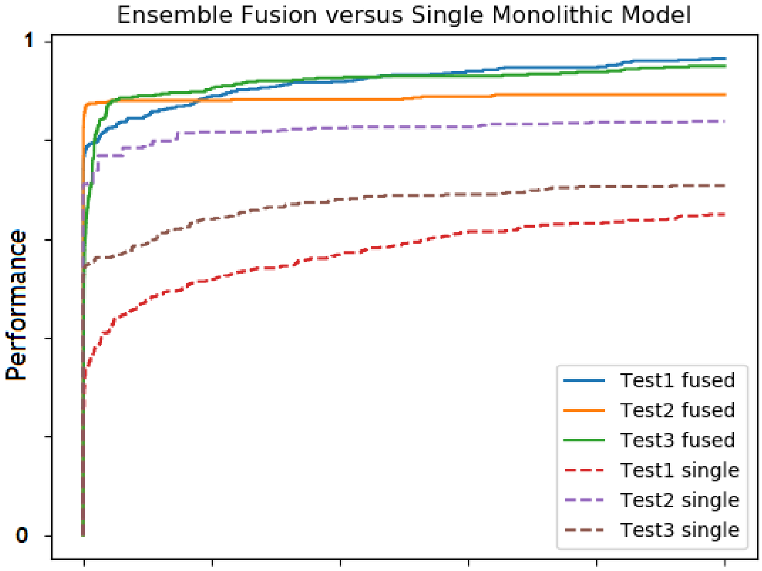


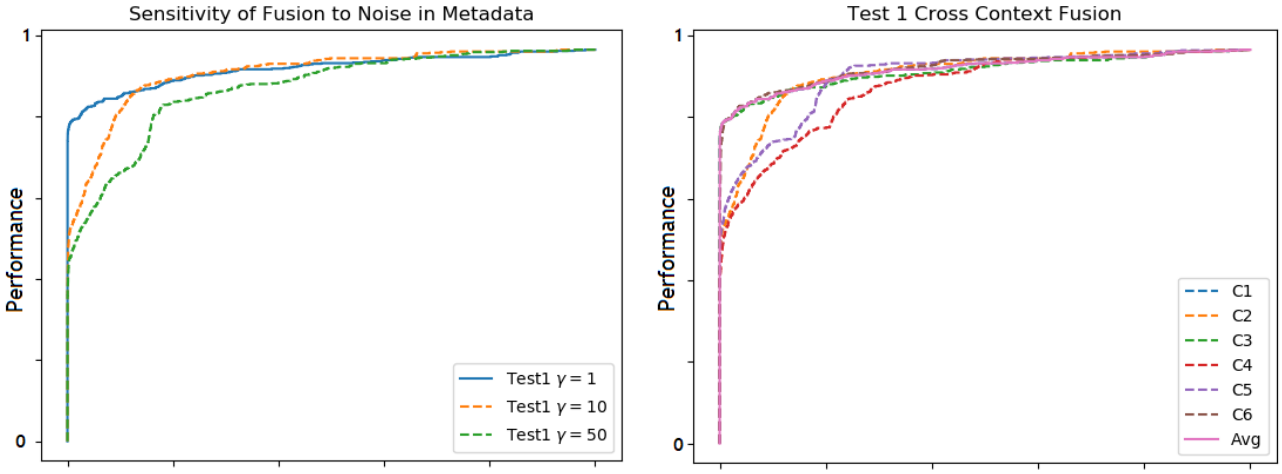
Figure 7: The proposed adaptive fusion scenario compared to a basic YOLOv5 architecture on three test scenarios. Test sets comprised of seen and unseen contexts.

versus plateaus (no detections but more false alarms) or "linear climbs" (aka you have a 50% chance of calling something target or a false alarm). Thus, on the three test contexts evaluated, our ensemble method performs better than the standard model in all cases. This is similar to what we observed relative to fusing, with a fixed versus adaptive strategy, a set of heterogeneous neural networks for land classification and object detection in remote sensing.[7,12–14] It should be noted that the total amount of data to train the ensemble is the same as the single model, though the single model was responsible for learning solutions across all of that data, while the ensemble was free to optimize a smaller subset of that data. While unproven, we believe this experimentally highlights the need to strike a balance between generalizeable models that perform well on all sorts of data and models that are experts in a more limited domain.

3.2 Sensitivity to Noise in Metadata

The above experiment is useful in the regard that it helps us understand operation in ideal scenarios. However, our method is reliant on additional data (metadata) provided by the UAS platform and/or environmental metadata. A benefit of using simulation is that we have complete control over the fidelity of this data. That is, we can simulate noise and other errors, which are likely to appear when the algorithm is used in the real world. To better understand the robustness of our method to such errors, we construct the folowing two sensitivity experiments.

Figure 8a depicts the degradation of fusion performance as noise is introduced to the associated metadata. Again, we are not disclosing which metadata (altitude, time of day, etc.) lead to the biggest degradation due to the sensitive nature of EHD. Specifically, our metadata was generated based on normal distributions with varying levels of standard deviation. The gamma variables in figure 8a are scalar multipliers to the base standard deviation, resulting in more erratic (and less representative) metadata. Thus, $\gamma = 1$ is a single standard deviation (normal operating conditions), $\gamma = 10$ is 10 and $\gamma = 50$ is 50 times more noise, respectively. Semantically, the simulated types of errors lead to incorrect identification of current contexts, or relying on the generated default operator. It can be noted that the training clusters present in the synthetic experiments are nicely separable and spaced apart. It is unclear (future work) how these algorithms will work in the case of extremely close contexts. If the metadata is a good context identification scheme then contexts would be expected to be distinct and separate in space. However, if context, and or collected data are very close, e.g., model for 1pm and another model for 1:30pm, then we might expect that the trained classifiers and fusions should be similar. The point is, further analytical studies with performance characterization or experiments need to be performed in order to understand the impact of adaptive fusion for such scenarios. In summary, this experiment (Figure 8a) informs



(a) Noise induced in metadata (b) All fusions compared on a single test run

Figure 8: Accurately capturing context data is important for the performance of our system. Excessive noise in the data can lead to incorrect operator selection as seen in 8a. The entire range of learned operators can also be evaluated on a single test run such as in 8b.

us that there is indeed an impact, but if our simulation is a close model to the operating conditions of a UAS for EHD, metadata noise is a concern but detection is not significantly impacted; we still get 40% detection with no error and do better than chance from that point on.

In Figure 8b, our second experiment, we consider what happens when the selection process consistently identifies an incorrect context. Whereas the last experiment showed us random controlled variations, this experiment tests extreme cases. This experiment is achieved by comparing the correct fusion operation to the rest of operators generated in the training process. These experiments results in six unique contexts being identified, thus, we can perform the fusion operation tailored to each of the six contexts. It should be no surprise that when the system is intentionally given incorrect contextual information, the performance of the fused algorithm is lowered, e.g., a morning model is used to detect EHs later in the day at a higher altitude. In general, we can see that the correct fusion operator (the generated weighted average) results in the best possible performance, though a few of the existing operators manage to perform equally well.

3.3 Explainability of Fusion

In this section we explore if there are any additional benefits of the proposed methodologies. One our underlying goals was to realize a trustworthy system versus opaque model. Our aim is to make a mathematical system that can be extended in future work, e.g., factor in additional advanced EHD or physics knowledge. The system outlined herein is really "level 1 visual intelligence". That is, the system processes imagery and tries to learn robust low-level spatial and spectral features. Ideally, the system outlined herein is not the entire system, but one stage in the detection and understanding pipeline. The end goal being a "higher level environmental understanding" algorithm. To that end, one of the primary benefits of using the proposed ChI fusion strategy, as opposed to a black-box neural network is that the ChI can be opened up and examined after the fact to determine what fusion strategies were prevalent. That is, the ChI is a centralized and explicit model versus a distributed and implicit neural network. However, we remark that in Ref. 11 we put forth a way to encode and optimize a full ChI as a neural network, without loosing interoperability. In this section we take a look at what was learned in the previous experiments. We only report a subset of explanations. The reader can refer to our past works[7,8,11,25] for our wider set of XAI fusion tools that generate statistical, graphical, local, and linguistic explenations.

In Murray et al.,[7,8] data-centric indices were proposed as a way of evaluating the kind and quality of data that was used to train the ChI. Of particular note is the walk-visitation calculation which describes what part of the Hasse diagram is well supported by data, i.e., how many (and which) internal contexts has a trained ChI observed and approximated. This can be used to identify "missing data", or perhaps more appropriately labeled "missing model variables." The trends reported herein are consistent across contexts, therefore we select and focus on the arbitrary Context 1. In this context, only 19% of the total possible walks received even a single piece of support. With this being a six source fusion, $6! = 720$ sorts are possible on the data, though only 137 of those were seen. Therefore, the integral is only approximately 20% approximated, which is not good. However, as we discuss in our fusion for remote sensing work,[7, 8, 12, 13] most real world datasets do not have sufficient volume nor diveristy. While many datasets claim to have both, our prior work showed that even for the higher volume datasets, their estimated fusion model values are frequently less than 20%. Meaning, our simulated scenario has arguably more diversity than we would encounter in practice. This makes sense to us, as the Unreal Engine lets us produce more data across different context.

Furthermore, of the walks that were taken, 65% of the time the data took the sort (1, 2, 3, 4, 6, 5), meaning that one walk almost completely dominates the operator. This is a common thing to encounter when training a ChI. That is, this is the default sort order. Usually that means that a bulk of the data is in agreement, is all saying the same thing. This is a typical behavior of strong learners, which we might expect from the YOLOv5 algorithm. Furthermore, this most prevalent walk corresponds to a **max** operator, meaning this particular operator tends to be optimistic. This is a difficult run in the Hasse to interpret. There is an fundamental entanglement that we cannot break apart. That is, this run is both the default sort order run and a valid case of what happens when algorithm 1 is more confident than algorithm 2, followed by 3, and so forth. As such, did we need the max to solve the latter or did it simply pick the max because it was an arbitrary selection when all algorithms say the same thing. Furthermore, the densities (values of the lowest level in the Hasse diagram) are $g(\{x_1\}) = 0.99, g(\{x_2\}) = 0.07, g(\{x_3\}) = 0.99, g(\{x_4\}) = 0.77, g(\{x_5\}) = 0.48, g(\{x_6\}) = 0$, showing that the fusion is often based entirely on the largest confidence value, as long as the largest confidence does not come from source two or six. Further analysis[7, 8, 11, 25] would be required to separate these variables to determine which are supported by data and we should trust.

This trend continues for most of the other contexts. Due to specific algorithms performing very well in each context (as the training procedure is therefore resubstitution), the fusion operators in those contexts weight those sources heavily. In future work we will look to sample and study validation data to minimize this effect. However, this is not the case for the default operator that was learned, as it was exposed to all data and it did not perceive a clear superior in the sources. As described in section 2.5, we explored multiple methods to generate default operators including an average aggregation and retraining on all data. While it would be difficult to display the full Hasse diagrams here (visually and with respect to page count), the operator that was retrained on all data resembles a **min** operator, while the average aggregation resembles a **mean**. This means that it is nearly impossible for the retrained operator to produce a high output, as all six algorithms would need to agree on a detection with a high confidence (something that rarely happens.) While it may seem a bit counter-intuitive to do all of this machine learning only to end up with something similar to what could be guessed at from the start (using a mean to aggregate sources), we believe that encouraging a less pessimistic model generalizes better to unseen data. The reader needs to keep a few things in mind. First, this is not conclusive and it is not a proof. It is merely an observed behavior of our experiments and experimental setup; i.e., simulated scenes, trained YOLOv5 classifiers, quadratic solver, etc. Thinking beyond our experiments, it is reasonable to expect that a high quality model trained for a specific context could prove to be *optimal*; versus the unachievable single model trained on all possible data or its ensemble approximation. Furthermore, we might expect that a mean like operator is an, on average, least worse strategy for outlier metadata scenarios. Last, when contexts truly overlap, something not explored yet, a mixture of models could prove to be a more robust approximation; similar to the performance gain we observed using fusion in Figure 7. Last, if we assume that each of these models and fusions are derived at least in part from data, then the models will always fundamentally have missing pieces. The point is, the above conclusion from our papers experiments are in no way conclusive. They are an experimental observation that we can use as intuition to set up the next and better approach.

4. CONCLUSION AND FUTURE WORK

This article proposes a metadata enabled adaptive fusion scheme for UAS-based EHD which attempts to discover the underlying contexts data was collected in to better inform the fusion operation. The offline determination of what constitutes a context is made by the possibilistic c-means (PCM) algorithm, a clustering algorithm which provides typicality values that describe to what degree a data point is typical of a given cluster. Once the training metadata is clustered, a set of context specific YOLOv5 location and detection classifiers are built, one per cluster. Finally, a Choquet integral (ChI) aggregation operator is trained for each context.

At evaluation time, the typicality values provided by the PCM allows the system to make an intelligent decision of whether or not an appropriate fusion scheme has already been trained. If the new data is sufficiently similar to an existing context then we are able to use the associated operator directly. If the new data is highly atypical from all previous contexts or similar to multiple contexts then the system uses a default strategy or it creates a new fusion scheme on the fly based on weighted interpolations of existing operators.

We evaluated the above methods on a set of synthetic imagery generated in the Unreal Engine, a process which allows us to circumvent the otherwise tedious process of obtaining large amounts of varied UAS data. Our results showed that there is benefit across the board in taking a metadata driven ensemble of our context dependent classifiers. Furthermore, we showed that while our system, as expected, is sensitive to metadata perturbation, the resultant ROC curve performance is still encouraging. Last, we showed additional sensitivity analysis experiments where we intentionally tried to destroy the algorithm. We note that this scenario is rare and might never be encountered in practice. However, the experiment reinforced our expected behavior of the system. That is, when out of context classifiers are used, performance is not ideal. However, our metadata fused result remains resilient. In summary, these preliminary experiments are encouraging.

In the future, we will evaluate how well simulator informed models transfer to real environments. We will apply intuition developed through our setup and experiments to an in depth investigation for each component in a real UAS scenario, e.g., metadata, its similarity, context prediction, etc. for GPS, IMU, and environmental factors. Furthermore, we only achieved a first step of adaptive fusion herein. That is, we use clustering to inform the construction of an on the fly fusion operator. In future work we will advance this model to include the factors discussed above, like internal context and its degree of approximation in a ChI for a given context. We want to advance this adaptive fusion method to let us produce operators that are similar to prior operators when we understand their performance. This may involve transferring solutions in and across models and metadata clustering typicality. Our goals are to determine if a well-trained model in a context outperforms an ensemble and to find optimal operators for addressing outliers. We intend to move away from experimentation and to rely on analytical proofs when possible to achieve this end. While preliminary experiments and the method are encouraging, there remains a great deal of future work.

5. ACKNOWLEDGEMENTS

Research funded by ARO grant number W911NF1810153 to support the U.S. Army DEVCOM C5ISR Center.

REFERENCES

[1] Anderson, D. T., Stone, K. E., Keller, J. M., and Spain, C. J., "Combination of anomaly algorithms and image features for explosive hazard detection in forward looking infrared imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **5**(1), 313–323 (2012).

[2] Gader, P. D., Mystkowski, M., and Yunxin Zhao, "Landmine detection with ground penetrating radar using hidden markov models," *IEEE Transactions on Geoscience and Remote Sensing* **39**(6), 1231–1244 (2001).

[3] Dowdy, J., Brockner, B., Anderson, D. T., Williams, K., Luke, R. H., and Sheen, D., "Voxel-space radar signal processing for side attack explosive ballistic detection," in [*Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXII*], Bishop, S. S. and Isaacs, J. C., eds., **10182**, 421 – 435, International Society for Optics and Photonics, SPIE (2017).

[4] Abdallah, A. C. B., Frigui, H., and Gader, P., "Adaptive local fusion with fuzzy integrals," *IEEE Transactions on Fuzzy Systems* **20**(5), 849–864 (2012).

[5] Murray, B., Islam, M. A., Pinar, A., Anderson, D., Scott, G., Havens, T., Petry, F., and Elmore, P., "Transfer learning for the choquet integral," 1–6 (06 2019).

[6] Kakula, S. K., Pinar, A., Islam, M. A., Anderson, D., and Havens, T., "Novel regularization for learning the fuzzy choquet integral with limited training data," *IEEE Transactions on Fuzzy Systems* , 1–1 (2020).

[7] Murray, B., Islam, M. A., Pinar, A., Anderson, D., Scott, G., Havens, T., and Keller, J., "Explainable ai for the choquet integral," *IEEE Transactions on Emerging Topics in Computational Intelligence* **PP**, 1–10 (07 2020).

[8] Murray, B., Islam, M. A., Pinar, A. J., Havens, T. C., Anderson, D. T., and Scott, G., "Explainable ai for understanding decisions and data-driven optimization of the choquet integral," in [*2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*], 1–8 (2018).

[9] Epic Games, "Unreal engine."

[10] Pinar, A. J., Havens, T. C., Islam, M. A., and Anderson, D. T., "Visualization and learning of the choquet integral with limited training data," in [*2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*], 1–6 (2017).

[11] Islam, M., Anderson, D. T., Pinar, A. J., Havens, T. C., Scott, G., and Keller, J. M., "Enabling explainable fusion in deep learning with fuzzy integral neural networks," *IEEE Transactions on Fuzzy Systems* **28**(7), 1291–1300 (2020).

[12] Scott, G. J., Hagan, K. C., Marcum, R. A., Hurt, J. A., Anderson, D. T., and Davis, C. H., "Enhanced fusion of deep neural networks for classification of benchmark high-resolution image data sets," *IEEE Geoscience and Remote Sensing Letters* **15**(9), 1451–1455 (2018).

[13] Scott, G. J., Hurt, J. A., Marcum, R. A., Anderson, D. T., and Davis, C. H., "Aggregating deep convolutional neural network scans of broad-area high-resolution remote sensing imagery," in [*IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*], 665–668 (2018).

[14] Scott, G. J., England, M. R., Starms, W. A., Marcum, R. A., and Davis, C. H., "Training deep convolutional neural networks for land–cover classification of high-resolution imagery," *IEEE Geoscience and Remote Sensing Letters* **14**(4), 549–553 (2017).

[15] Narukawa, Y. and Murofushi, T., [*Choquet integral and Sugeno integral as aggregation functions*], 27–39, Springer Berlin Heidelberg, Berlin, Heidelberg (2003).

[16] Islam, M. A., Anderson, D. T., Pinar, A. J., and Havens, T. C., "Data-driven compression and efficient learning of the choquet integral," *IEEE Transactions on Fuzzy Systems* **26**(4), 1908–1922 (2018).

[17] Islam, M. A., Anderson, D. T., Petry, F., and Elmore, P., "An efficient evolutionary algorithm to optimize the choquet integral," *International Journal of Intelligent Systems* **34**(3), 366–385 (2019).

[18] Krishnapuram, R. and Keller, J. M., "The possibilistic c-means algorithm: insights and recommendations," *IEEE Transactions on Fuzzy Systems* **4**(3), 385–393 (1996).

[19] Ross, T. J., [*Fuzzy C-Means Algorithm*], 358, Wiley (1995).

[20] Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., and Bailey, J., "Online cluster validity indices for performance monitoring of streaming data clustering," *International Journal of Intelligent Systems* **34**(4), 541–563 (2019).

[21] Islam, M. A., Anderson, D., Petry, F., and Elmore, P., "An efficient evolutionary algorithm to optimize the choquet integral," *International Journal of Intelligent Systems* **34** (09 2018).

[22] Yager, R. R., "A measure based approach to the fusion of possibilistic and probabilistic uncertainty," *Fuzzy Optimization and Decision Making* **10**, 91–113 (June 2011).

[23] Jocher, G., Stoken, A., Borovec, J., NanoCode012, ChristopherSTAN, Changyu, L., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang1900, Chaurasia, A., Diaconu, L., Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, Ingham, F., Frederik, Guilhen, Colmagro, A., Ye, H., Jacobsolawetz, Poznanski, J., Fang, J., Kim, J., Doan, K., and Yu, L., "ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration," (Jan. 2021).

[24] van der Maaten, L. and Hinton, G., "Viualizing data using t-sne," *Journal of Machine Learning Research* **9**, 2579–2605 (11 2008).

[25] Murray, B. J., Anderson, D. T., Havens, T. C., Wilkin, T., and Wilbik, A., "Information fusion-2-text: Explainable aggregation via linguistic protoforms," in [*Information Processing and Management of Uncertainty in Knowledge-Based Systems*], Lesot, M.-J., Vieira, S., Reformat, M. Z., Carvalho, J. P., Wilbik, A., Bouchon-Meunier, B., and Yager, R. R., eds., 114–127, Springer International Publishing, Cham (2020).