Multimodal Data Fusion Using Canonical Variates Analysis Confusion Matrix Fusion

Erik Blasch
Air Force Reseach Lab
Information Directorate
Rome, NY, USA

Asad Vakil
Electrical and Computer Eng.
Oakland University
Rochester, MI48309
avakil@oakland.edu

Jia Li
Electrical and Computer Eng.
Oakland University
Rochester, MI48309
li4@oakland.edu

Robert Ewing
Air Force Research Lab
Sensors Directorate
WPAFB. OH, USA

***Abstract*—*Data fusion from a variety of sources requires alignment, association, and analysis. One method to determine the relationship between two variables measuring the same information is a correlation analysis. The canonical variates analysis (CVA) supports the assessments of two sets of data. In this paper, we compare results from the fusion of histograms to that of the fusion of confusion matrices developed from data of the same modality and that of a cross modality. We use the Confusion Matrix Fusion (CMF) approach in the analysis and compare the results for EO/RF fusion. In the analysis, the Experiments, Scenarios, Concept of Operations, and Prototype Engineering (ESCAPE) data set is used for comparison to previous aerospace results.*

Keywords: *Canonical Variates Analysis, information fusion, veracity, confusion matrix fusion*

TABLE OF CONTENTS

1. INTRODUCTION

Data fusion from electro-optical (EO) and radio frequency (RF) target classification extends operations for all weather day-night analysis [1,2,3,4].EO/RF fusion has a wide variety of applications including target surveillance, agricultural monitoring, and safety from UAV [5], space [6], and automotive [7] platforms. The development of antennas to process EO and RF data [8,9] has renewed interest for autonomous cars and aircraft [10], offering possibilities for enhanced situation awareness and decision support [11].

Recent efforts with the *Experiments, Scenarios, Concept of Operations, and Prototype Engineering* (ESCAPE) dataset [12], has explored many techniques for analysis with an example shown in Fig. 1. Various examples include nonparametric Bayesian fusion, joint manifold fusion, and structural similarity [13,14,15,16,17,18].

The scenario in Figure 1 highlights the motivation to track targets using a series of measurements from electro-optical (EO), infrared (IR), passive radio frequency (P-RF), full motion video (FMV), seismic, and acoustic data.

Recent advances in artificial intelligence and machine learning have been developed, but mostly on single-modal situations (EO-IR) or multi-modal in which there is a known association between the data sources such as in video and audio multimedia. Hence, there is an opportunity to determine the merits of AI/ML techniques [19,20,21]. Recently, the convolutional neural network (CNN) has been applied to the ESCAPE data with some success; however, elements of data fusion can be further expanded. More specifically, the known paradigm of signals, feature, and decision-level fusion offer the question on which approach works well for the sensor, environment, and target (SET) operating conditions. Since the data is collected with the different sensors, decision-level fusion was utilized as a comparison.

To address the comparisons of machine learning methods, there are various statistical approaches such as the nearest neighbor, k-nearest neighbor (kNN), decision tree, and naive Bayes, as well as deep learning approaches such as the convolution neural network (CNN). A fusion of classifiers approach might be an option, but two methods of known success include confusion-matrix fusion (CMF) and correlation variates analysis (CVA), also known as canonical correlation analysis (CCA) [22,23,24,25,26]. Hence, a decision-level fusion approach is compared to the methods of deep learning.

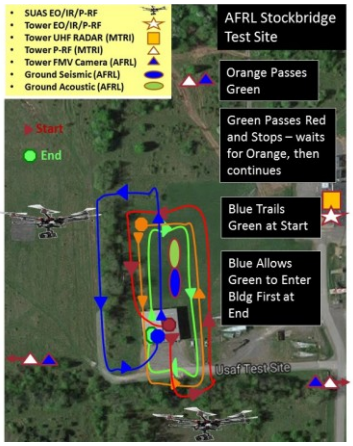

Figure 1. ESCAPE data collection

The paper is organized as follows. Sect. 2 presents the ESCAPE data, while Sect. 3 overviews data fusion classification methods. Sect 4 presents confusion matrix fusion as a method of decision-matrix fusion. Sect. 5 presents some numerical results. Finally, conclusions are drawn in Sect.6.

2. ESCAPE DATA

The ESCAPE payload is a collection of three sensing modalities, EO FMV, IR FMV and passive RF (EO/IR/P-RF). Two Unmanned Aerial Systems (UAS) contain EO/IR/P-RF sensors on a Vertical Take Off and Landing

2021 IEEE Aerospace Conference | U.S. Government work not protected by U.S. copyright | DOI: 10.1109/AERO50100.2021.9438445

(VTOL) aircraft. To monitor the events, another EO/IR/P-RF was mounted on a tower (Figure 2).

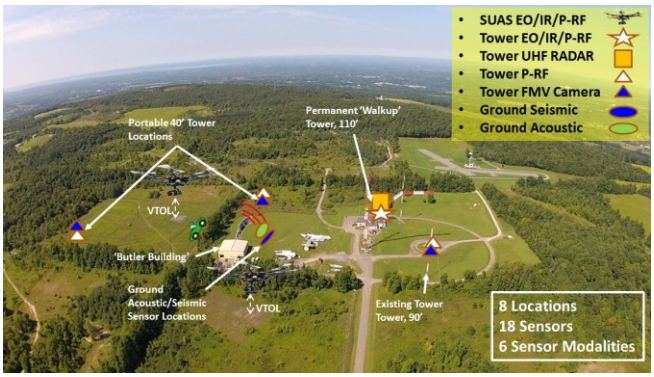

Figure 2. ESCAPE Data

Radar Returns

Figure 3 displays the UHF radar returns for the multi-vehicle scenario. The right side of the Figure 3 displays amplitude (color code) versus range (vertical axis) versus time (horizontal axis). As indicated, certain paths, predictably, have different *traces*. For example, the orange path vehicle, panel van, starts in view of the radar and moves away from the radar until it is lost in the Butler building, as indicated in the radar return. The wide band of returns between 160 m and 200 m represents the Butler building and tree line scattering, and hence, the van return gets consumed by this clutter as it is blocked from the radar. Later in time, the van re-emerges from the Butler building and is easily detectable in the diagram between pulse indexes 800 and 1400.

The data provides an opportunity for more interesting fusion approaches to make use of the radar phase-history information in order to assign each target a unique feature that correlates with other modality specific signatures.

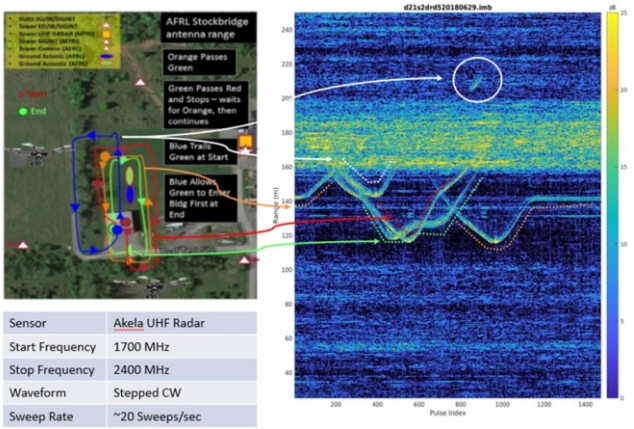

Figure 3. UHF radar returns (right side) of a multi-vehicle scenario. Returns are in range (vertical axis) vs. time (horizontal axis). Colored arrows relate the vehicle paths with the radar returns.

EO and IR Association

Figure 4 displays a snapshot in time from the recorded video data from 3 of the ESCAPE payloads, positioned as shown in Figure 2. In Figure 4, row (a) is from the 110' foot walk up tower, row (b) is from the most Westerly SUAS position, and row (c) is from the most Southerly SUAS position. Interestingly, some of the moving vehicles exist in most of the cameras simultaneously.

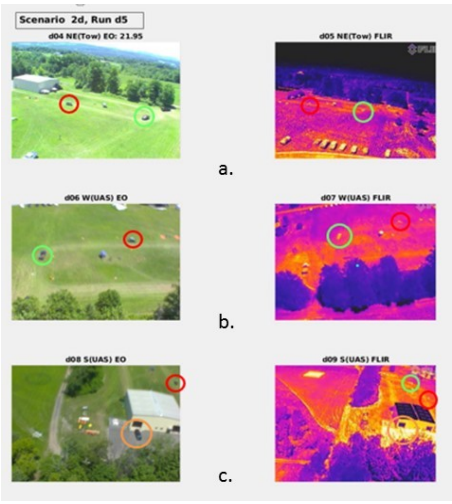

Figure 4. ESCAPE payload EO and IR frames during a data collection run of a scenario (4 vehicles). (a) 110' tower view, (b) Westerly SUAS view, (c) Southerly SUAS view. Color code: red-Gator, green – pickup, orange – van.

The objective of the research in this paper and many other papers studying the ESCAPE data set will be to determine if a joint representation of each target can be produced that is unique to the given target in order to accurately and optimally improve target detection, improve discriminability, and improve target tracking. With overlapping fields of view across the three camera locations and across the EO and IR modalities, image fusion [27, 28] could yield robust results.

3. Data-Fusion Classification

Data fusion classification from similar modes (EO/IR) has the benefit of similar types of sensors. Multi-modal techniques include that of similar format such as 2D imagery from EO and synthetic aperture radar (SAR) imagery [29] as well as that of different types of modes such as 1D radar and 2D imagery [30,31]. As shown in Figure 5, there are different constructs when processing the EO dictionary of feaures and RF historgams. Processing the mulitmodal data can serve to reduce the cross-covariance [32] and cue detection [33]. To further explore the potential for multimodal EO/IR and radar fusion, we expand upon machine learning methods [34]. The next section descirbes the machine learning techniques used for comparison to proces the data towards a confusion matrix (CM).

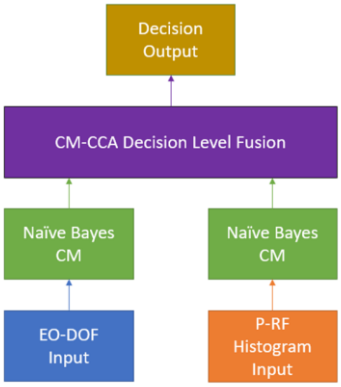

Figure 5. Data Fusion Analysis

K-nearest Neighbor (KNN)

The K-nearest Neighbor (KNN) is a non-parametric method (i.e., distribution free) of instance learning that works on entire data set at once to discern a pattern in feature space that determines the k closest examples. K is a specified positive integer, which is usually relatively small. The KNN does not learn any model – rather the model is itself the training set. For each new instance, the algorithm searches through the entire training set, calculating the difference between the new instance and each training model. A classic example is the Voronoi diagram (Figure 6) which partitions regions based on a distance to the features. Once the distance regions are determined, then a corresponding boundary set is available for classification.

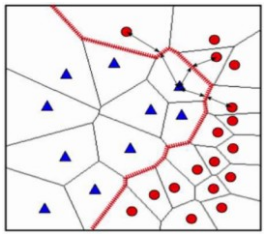

Figure 6. K-nearest neighbor from the Voronoi diagram

- For ***classification***, the output is the class with the K-most similar neighbors.
- For ***regression***, the output value is based on the mean or median of the K-most similar instances

Both for classification and regression, the kNN assigns weights to the features, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of $1/d$, where d is the distance to the neighbor. An example is shown in Figure 7. The test sample (green dot,) should be classified either to blue triangles or to red circles. If k = 3 (solid line circle), it is assigned to the blue triangles ( ), because there are 3triangles and no red circles inside the inner circle. If k = 20 (dashed line circle), it is assigned to the red circles ( ), as there are 11 circles vs. 9 triangles inside the outer circle.

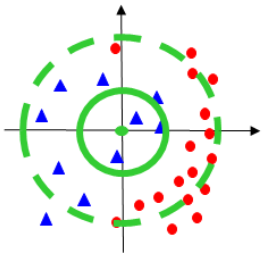

Figure 7. Example of k-NN classification.

The kNN works to classify the training data point x_q (green in the above diagram).

- Given a query instance q to be classified
  - Let $x_1, \ldots, x_k$ be the k training instances in training set $\mathbf{T} = (x, f(x_i))$ nearest to q
  - Return

$$\hat{f}(q) = \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i)) \qquad (1)$$

- where v is the finite set of target class values, and $\delta(a,b)=1$ if $a=b$, and 0 otherwise (Kronecker function)
- then, the k-NN algorithm assigns to each new query instance the majority class among its k nearest neighbors

A distance weighted kNN uses a difference between the test point x_q and the training data x_i :

$$\hat{f}(q) = \arg\max_{v \in V} \sum_{i=1}^{k} \frac{1}{d(x_i, x_q)^2} \delta(v, f(x_i)) \qquad (2)$$

There are many distance functions, such as the city block (Manhattan), Chebyshev, Minkowski, quadratic, correlation, and Chi-square. Two popular methods include:

Euclidian distance:

$$d(x_i, x_q) = \sqrt{(x_i - x_q)^2 + \cdots + (x_n - x_q)^2}$$

Mahalanobis:

$$d(x_i, x_q) = [\det V]^{(1/m)} (x_i - x_q)^T V^{-1} (x_i - x_q)$$

where V is the covariance matrix, *det* is the determinant and T is the transpose operator.

A practical method is estimating the density at a point x as the reciprocal of the average of the distances to the K nearest neighbors of x:

$$f(x) = \frac{1}{\frac{1}{K}\sum_{n \in N} d(x, n)} \qquad (3)$$

where N denotes the K-nearest neighbors of x, and n is an

element of N. Note that increasing the value of "K" takes more information about a point into consideration. Setting K=1, the estimate is completely local, which will result in producing a high number of clusters. Larger values of K will yield increasingly global estimates, while decreasing the granularity of the result, so that fewer clusters are produced. KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance. In general, the KNN is similar to the SVM with a Gaussian kernel.

Benefits of KNN

- Applicable — no assumptions about data distributions
- Simple — easy to explain and understand/interpret
- Accurate — achieves reasonable results
- Versatile — useful for classification or regression

Limitations of KNN

- Computationally expensive by storing all data
- Slow - prediction takes time (with big N)
- High memory requirement so does not scale
- Susceptible to irrelevant features
- Sensitive to the local structure of the data

Decision Tree

The random forest (RF) approach is an ensemble learning method that derives its name from splitting decision trees where oblique hyperplanes gain accuracy as they grow without suffering from overtraining. If the splitting method is randomly forced to be insensitive to some feature dimensions or randomly restricted to be sensitive to only selected feature dimensions, then precision is gained with each tree split in the forest. Note that when decision trees are designed to be deep, they learn irregular patterns and overfit (i.e., low bias, high variance) the data. The RF output is the mode of classes for classification or mean of individual trees for regressions. RF(s) average multiple deep decision trees using the same training set reducing the variance with small increase in bias and boosting the final performance. Constructing a forest of uncorrelated trees extends a Classification and regression Trees (CART) decision-tree like procedure with randomized node optimization and bagging. Bagging is a method of boostrap aggregation to reduce variance through averaging. Hence, bootstrap sampling is a way of de-correlating the trees by showing them different training sets. RF estimates the generalization error, measures variable importance through permutation, and assesses the tree correlations (i.e., want trees to be uncorrelated).

There are three types:

- **Bagging:** train learners in parallel on different samples of the data, then combine by voting (discrete output) or by averaging (continuous output).
- **Stacking:** combine model outputs using a second-stage learner like linear regression.
- **Boosting:** train learners on the filtered output of other learners.

Given a training set $X = \{x_1, ..., x_i\}$ with responses $Y = \{y_1, ..., y_i\}$,the bagging type of RF repeatedly (N times) selects a random sample with replacement from the training set and fits trees to these samples:

For n = 1, ..., N:

1. Sample, with replacement, n training examples from X, Y; for X_n, Y_n.
2. Train a classification or regression tree h_n on X_n, Y_n.

After training with $X_n = \{x_1, ..., x_n\}$, predictions for unseen samples $X_u = \{x_1, ..., x_u\}$, where $X_n + X_u = X$, are:

$$\hat{h}(x_u) = \frac{1}{N}\sum_{i=1}^{N} h_i(x_u) \tag{4}$$

or by taking the majority vote in the case of classification trees.

Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees onX_u:

$$\hat{\sigma}(x_u) = \sqrt{\frac{\sum_{i=1}^{N}\left(h_i(x_u) - \hat{h}(x_u)\right)^2}{N-1}} \tag{5}$$

The number of samples/trees, N, is a free parameter. Typically, 100s to 1000s trees are used, depending on the size of the training set. An optimal number of trees N can be found by using cross-validation, or via the *out-of-bag error:* the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample. A Random Forest tree is shown in Figure 8 with N = 3.

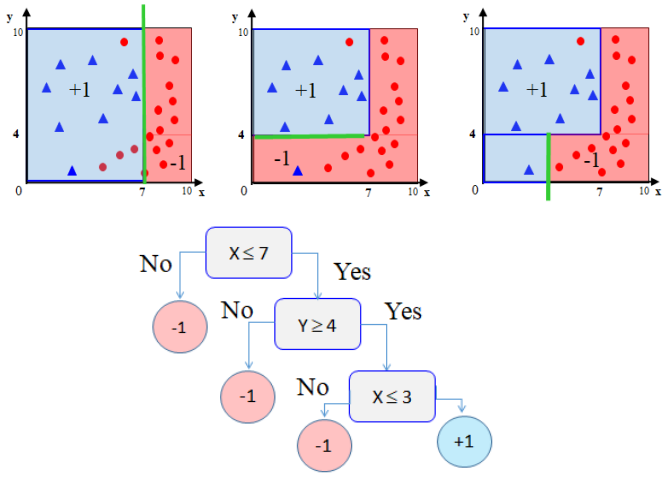

Figure 8. Decision Tree diagram from classification partitions

The RF procedure is as follow.

A. Grow n trees on datasets sampled from the original dataset with replacement (bootstrap samples), F = number of features.

- Draw n bootstrap samples of size N

- Grow each Decision Tree, by selecting a random set of m out of F features at each node, and choosing the best feature to split on where it is suggested that $\sqrt{f}$ (rounded down) features are used in each split
- Aggregate the predictions of the trees (most popular vote) to produce the final class.

B. Each tree is constructed using the following algorithm:

Draw n bootstrap samples of size N and the number of variables in the classifier be M.

1. Let m input variables determine the decision at a node of the tree; $m \leq M$.
2. Grow each Decision Tree by choosing a training set for this tree by choosing n times with replacement from all N available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
3. For each node of the tree, *randomly* choose m variables (i.e., feature bagging) on which to base the decision at that node. Calculate the best split based on these m variables in the training set (e.g., $\sqrt{m}$)
4. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier) where prediction a new sample is pushed down the tree and assigned the label of the training sample in the terminal node it ends up in.
5. Iterated over all trees in the ensemble
6. Aggregate the predictions of the trees (most popular vote) to produce the final class.

To additional considerations:

Extremely randomized trees, or ExtraTrees, add one further step of randomization. Using an ensemble of individual trees, each tree is trained using the whole learning sample (rather than a bootstrap sample), and top-down randomized splitting. Hence, instead of computing the locally *optimal* cut-point for each feature under consideration (based on, e.g., information gain), a *random* cut-point is from a uniform distribution within the feature's empirical range (in the tree's training set). Then, from all ofthe randomly generated splits, the split that yields the highest score is chosen to split the node.

Importance of variables used as a ranking for regression or classification. During the fitting process of the data set $\mathcal{D}_n=\{(x_i, y_i)\}_{i=1}^n$, the *out-of-bag error* for each data point is recorded and averaged over the forest. The importance score for the j-th feature is computed by averaging the difference in out-of-bag error before and after the permutation over all trees. The score is normalized by the standard deviation of these differences. Features which produce large values for this score are ranked as more important than features which produce small values.

As part of their construction, random forests can be used for semi-supervised methods by defining a *decision tree dissimilarity measure* between unlabeled data: by constructing a random forest predictor that distinguishes the "observed" data from suitably "generated" synthetic data. The observed data are the original unlabeled data and the synthetic data are drawn from a reference distribution. A random forest dissimilarity affords mixed variable types, is invariant to monotonic transformations of the input variables, is robust to outlying observations, and can weigh the contribution of each variable according to how dependent it is on other variables. The RF is essentially a weighted k-nearest neighbor (kNN) method and variations include a kernelized RF.

Benefits of Decision Tree

- Useful – applies to dense data (≤100 features)
- Simple– can train a lot of trees
- Efficient - parallelizes easily

Weakness of Decision Tree

- Biased - favors tree attributes with more levels.
- Timeliness - needs many passes over the data
- Overfits– hard to balance accuracy/fit tradeoff.

Naive Bayes

Naive Bayes develops a conditional probability classifier model of an instance C_k by assigning labels from a finite set to problem instances, represented as feature vectors $\mathbf{x} = (x_1, \ldots, x_n)$ of n independent feature values. Using Bayes' theorem:

$$p(C_k \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_k)\, p(C_k)}{p(\mathbf{x})} \tag{6}$$

where $p(C_k \mid \mathbf{x})$ is the posterior, $p(\mathbf{x} \mid C_k)$ is the likelihood, $p(C_k)$ is the prior, and $p(\mathbf{x})$ is the evidence. Naïve Bayes' is the conditional independence of the mutually independent features $\mathbf{x}$ for conditional on the category C_k.

$$p(C_k \mid x_1, \ldots, x_n) = \frac{p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)}{\sum_k p(C_k)\, p(\mathbf{x} \mid C_k)} \tag{7}$$

Using the maximum a posteriori (MAP) decision rule for the most probable choice, then the class selection label $\hat{y} = C_k$ for an instance k is:

$$\hat{y} = \underset{k \in \{1,\ldots K\}}{\operatorname{argmax}}\, p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k) \tag{8}$$

Benefits of Naïve Bayes

- Compact – requires few samples for training
- Simple – easy to implement
- Efficient – updates quickly with new data [35]

Weakness of Naïve Bayes

- Collection - assumes mutual independence (92%)
- Zero Frequency – requires training category in test set
- Overlap – decision boundaries need to be distinct

Canonical Variates Analysis

The CVA or Canonical Correlation Analysis (CCA), first described by Hotelling, works on a two data sets. As demonstrated by Sun [21], let the data be represented as X = $[x_1, \ldots, x_m]$ and Y = $[y_1, \ldots, y_m]$. CCA attempts to seek two projection directions w_x and w_y to maximize the following linear correlation coefficient:

$$\rho = \frac{cov(w_x^T X, w_y^T Y)}{\sqrt{var(w_x^T X) var(w_y^T Y)}} = \frac{w_x^T C_{xy} w_y}{\sqrt{(w_x^T C_{xx} w_x)(w_y^T C_{yy} w_y)}} \quad (9)$$

where the covariance matrix C_{xy} is defined as:

$$C_{xy} = \frac{1}{m}\sum_{i=1}^{m}(x_u - \mu_x)(x_u - \mu_x)^T \quad (10)$$

with μ_x and μ_y being the means from the two data sources and C_{xx} and C_{yy} can be defined analogously.

Since the scales of w_x and w_x have no effects on the solution, so each of the two factors in the denominator can be constrained to have value 1. Hence, an optimization can be formed using an objective function for the CCA

$$\arg\max_{w_x, w_y} w_x^T C_{xy} w_y$$
$$\text{s.t.} w_x^T C_{xx} w_y = 1;\ w_x^T C_{yy} w_y = 1 \quad (11)$$

Using a Lagrangian function approach, then

$$\lambda = w_x^T C_{xy} w_y \quad (12)$$

where $\lambda \in [-1,+1]$. reflects the degree of correlation between projections. Using the generalized eigenvalue decomposition, the eigenvector corresponding to the largest eigenvalue in (13) should be retained to maximize the correlation between different data sources.

$$C_{xy} C_{yy}^{-1} C_{yx} w_x = \lambda^2 C_{xx} w_x \quad (13)$$

The benefits and weakness of CCA/CVA is the motivation for the analysis. Given the different methods, the next section focuses on decision matrix fusion as related to a fusion of classifiers.

4. Decision-Matrix Fusion

Automatic target recognition (ATR) classifier decisions are often stored in a confusion matrix (CM), which is an estimate of likelihoods. For single-look ATR performance, these estimates are treated as priors. Decisions from multiple ATRs or from multiple looks of different geometric perspectives are fused using the Decision Level Fusion (DLF) technique. Various methods support the signal, feature, and decision fusion [36] depending on access to the CM for such applications as target tracking [37] .

With respect to the DLF, the CMs represent the performance of the ATR system from historical information. However, the performance is based on the availability of ground-truth or pseudo-ground truth. For example, in target tracking; pseudo-ground truth includes such actions as labeling targets in images or signatures, whereas ground truth is from the location instrumentation and verification to measurements. Hence, the historical performance of a sensor and classifier leads to a CM for an ATR. Choosing a different sensor and classifier result in a different CM [38]. For dynamic sensors, the CM can be used for sensor management and evidential tracking performance. When the sensor measurement is received, the evidential reasoner determines the probabilities across the potential targets for the CM. When new measurements are available, the new CM is determined and fused with the previous CM. For further details, the Dezert-Smarandache Theory (DSmT) [39] and Transferable Belief Model (TBM) [40] are widely used. Additional data can be gathered to utilize conditional information to improve the CM [41].

Assume that we have two ATRs each described in a confusion matrix designated as C^A and C^B. The elements of a confusion matrix are c_{ij} = Pr{ATR decides o_j when o_i is true}, where i is the true object class, j is the assigned object class, and i = 1, …., N for N true classes. The CM elements can be represented as probabilities as c_{ij}= Pr$\{z = j|o_i\}$ = $p\{z_j|o_i\}$. To determine an object declaration, we need to use Bayes' rule to obtain $p\{o_i|z_j\}$ which requires the class priors, $p\{o_i\}$. The priors and likelihoods are denoted as column vectors

$$p(\bar{o}) = \begin{bmatrix} p(o_1) \\ p(o_2) \\ \vdots \\ p(o_N) \end{bmatrix};\ p(z_j|\bar{o}) = \begin{bmatrix} p(z_j|o_1) \\ p(z_j|o_2) \\ \vdots \\ p(z_j|o_N) \end{bmatrix} \quad (14)$$

For M decisions, a confusion matrix would be of the form

$$C = \begin{bmatrix} p(z_1|o_1) & p(z_2|o_1) & .. & p(z_M|o_1) \\ p(z_1|o_2) & p(z_2|o_2) & .. & p(z_M|o_2) \\ \cdots & \cdots & \ddots & \cdots \\ p(z_1|o_N) & p(z_2|o_N) & .. & p(z_M|o_N) \end{bmatrix}. \quad (15)$$

The joint likelihoods are similar column vectors, where we assume independence for two confusion matrices A and B (denoted here as superscripts),

$$p(z_j^A, z_k^B|\bar{o}) = \begin{bmatrix} p(z_j^A|o_1) \bullet p(z_k^B|o_1) \\ p(z_j^A|o_2) \bullet p(z_k^B|o_2) \\ \cdots \\ p(z_j^A|o_N) \bullet p(z_k^B|o_N) \end{bmatrix}, \quad (16)$$

where k is used to distinguish between the different assigned object classes between the two confusion matrices when the CMs are not symmetric.

Using the priors and the likelihoods, we can calculate *a posteriori* from Bayes' Rule (where for purposes of clarification, $p(\bar{o}|\bullet)$ is used to denote a vector in the implementation of the code):

$$p(\bar{o}|z_j^A, z_k^B) = \frac{p(z_j^A, z_k^B|\bar{o}_j)p(\bar{o}_j)}{\sum_{i=1}^{n} p(z_j^A, z_k^B|\bar{o}_i)p(\bar{o}_i)} \quad (17)$$

Note that there are similar column matrices for the posteriors $p(\bar{o}|z_j)$ and $p(\bar{o}|z_j^A, z_k^B)$. A decision is made using the maximum likelihood estimate

$$d_i = \arg\max_{j,k} p(o_i|z_j^A, z_k^B) \quad (18)$$

where the final decision of the true object class i is determined from the largest value from the vector.

Note that the subscripts indicate the value of a variable and the superscripts indicate the ATR source. For example, $z^A = z_3$ indicates that source A made a decision z_3; where source A might be the first look of a HRR ATR and decision z_3 might be target type "sedan". The absence of a superscript implies an unspecified single source. To represent the particular states from each source, they are indicated with the subscripts a and b (such as $z^A = z_a^A$) indicating that source A's decision was z_a.

For the developments of the pseudo code, shown in Algorithm 1, we shorten the notation to $z^A = z_a$, while keeping track of the confusion matrix source A or B.

Naïve Bayes DLF Pseudocode

Inputs to the fuser are the decisions of ATR A and B, i.e., z_a and z_b respectively. The output of the fuser is the decision d based on a maximum a posteriori probability (MAP) decision rule, where the posterior is . The fuser must know the prior probabilities and the confusion matrices (one for each source).

Pseudo code for decision level fusion is represented as:

- za = z_a and zb = z_b are the integer decisions between 1 … M of sources A and B, respectively
- pObar = is a vector of priors, represented as either constants or input variable
- CA = C^A and CB = C^B are the confusion matrices derived from sources A and B, respectively
- pZaObar = and pZbObar = are the likelihoods as extracted columns from the confusion matrices [pZaObar = CA(:,za); and pZbObar = CB(:,zb)]
- pZaZbMbar = is the joint likelihood derived from the point-wise product of the source likelihoods (pZaZbObar = pZaObar .* pZbObar);
- pObarZaZb = $p(\bar{o}|z_a, z_b) = \frac{p(z_a, z_b|\bar{o})p(\bar{o})}{\sum_{i=1}^{n} p(z_a, z_b|\bar{o}_i)p(\bar{o}_i)}$
  - the numerator is:

    posteriorNum = pZaZbObar .* pObar;
  - the denominator is:

    posteriorDen = sum(posteriorNum);
  - pMbarZaZb = posteriorNum / posteriorDen;
- d = max(pObarZaZb), which is the fused decision,

  $d_i \ni p(o_i \mid z_a, z_b) \geq p(o_i \mid z_a, z_b)$

  $\forall i, j$ where $i, j \in 1, \ldots, N$

The DLF function pseudo code is presented for verification.

Algorithm 1: Confusion Matrix Fusion

```
function [d, pObarZaZb]=fuseCMdecisions(za, zb, Obar)
        CA = getConfusionMatrix(1);
        CB = getConfusionMatrix(2);
        pZaObar = CA(:,za);
        pZbObar = CB(:,zb);
        pZaZbObar = pZaObar.* pZbObar;
        posteriorNum = pZaZbObar .* pObar;
        posteriorDen = sum(posteriorNum);
        pObarZaZb = posteriorNum / posteriorDen;
        [junk, d] = argmax(pObarZaZb);
        Return
```

5. RESULTS

From the ESCAPE dataset, Scenario 2 was chosen as a baseline case. Table 1 presents the results from various machine learning methods for the classification of three vehicles. These results are the average of the classification over the scenario for the probability of detection. Likewise, classification confusion matrices were available for processing. Table 1 results show that the imagery provided a better classification, while one classifier was not superior to the rest in all cases. Hence, typically there are methods for fusion of classifiers.

Table 1: Machine Learning Classification results based on single modality input for the three vehicles in Scenario 2:

Classifier:	Naïve Bayes	Decision Tree	KNN	Nearest Centroid
V1 EO	**0.800**	0.756	0.780	0.799
V1 RF	0.505	0.562	0.572	**0.629**
V2 EO	0.787	0.756	0.781	**0.811**
V2 RF	0.518	0.562	0.572	**0.635**
V3 EO	0.627	**0.846**	0.509	0.613
V3 RF	0.543	0.612	0.580	**0.621**

For the results from the other two scenarios, Table 2 and Table 3 demonstrate similar results as seen below. There are some outliers within single modality input classification, such as Vehicle 1 in Scenario 3. The application of the KNN clustering is the only instance of the P-RF histogram input by itself achieving a performance of 0.9 in all three scenarios, let alone outperforming the EO data for the same vehicle and classifier.

Table 2: Machine Learning Classification results based on single modality input for the two vehicles (V) in Scenario 1:

Classifier:	Naïve Bayes	Decision Tree	KNN	Nearest Centroid
V1 EO	0.949	0.639	**0.962**	0.593
V1 RF	**0.640**	0.451	0.622	0.413
V2 EO	0.959	0.639	**0.963**	0.639
V2 RF	0.584	0.451	**0.783**	0.413

Table 3: Machine Learning Classification results based on single modality input for all five vehicles (V) in Scenario 3:

Classifier	Naïve Bayes	Decision Tree	KNN	Nearest Centroid
V1 EO	0.800	0.907	0.903	**0.912**
V1 RF	0.714	0.719	**0.905**	0.750
V2 EO	0.894	0.900	0.881	**0.917**
V2 RF	0.664	0.610	**0.794**	0.682
V3 EO	0.864	0.913	0.850	**0.928**
V3 RF	0.643	0.578	**0.736**	0.604
V4 EO	0.818	**0.917**	0.913	0.785
V4 RF	0.694	0.816	0.826	0.643
V5 EO	0.917	0.876	0.874	0.862
V5 RF	0.820	0.642	0.854	0.618

To illustrate the differences in performance between models that rely on only one modality in this dataset, Table 4 and 5 display the normalized confusion matrixes for models that only use P-RF and DOF-EO data. The detection of Vehicle 2 is extremely difficult given the limited amount of time it appears in the scenario and the reliability of the P-RF histograms and the DOF-EO video input. The P-RF histograms for the most part do not provide enough discriminative features to sufficiently match the classification objective across all three scenarios.

Table 4: Combined standalone EO Normalized Confusion Matrix (Accuracy 0.74) for Vehicle 2 in Scenario 2:

	Precision	Recall	F-1 Score
Target Not Detected	0.8	0.79	0.79
Target Detected	0.63	0.64	0.64

Table 5: Combined standalone RF Normalized Confusion Matrix (Accuracy 0.56)for Vehicle 2 in Scenario 2:

	Precision	Recall	F-1 Score
Target Not Detected	0.65	0.68	0.67
Target Detected	0.38	0.34	0.36

The advantages of the confusion matrix fusion compared to CNN Fusion and the standalone modalities can be seen below. For the CNN, it was an eight layer convolutional 2D of kernel size 3x3, which after max pooling, flattening, dropout, and dense layers [23] is then rerouted through a CCA using the pseudo layer, which evaluates that iteration of the data and then feeds the maximally correlated results back into the CNN.

Comparing the results of Tables 6 and 7 for the same vehicle in the same scenario, the increased F-1 score, and the improved reliability of the model shows a drastic increase in model performance. However, once the application of CM-CCA fusion is implemented the performance of the model achieves a perfect F-1 score, as seen in Table 8. The results in Table 8 are the results based on the CM-CCA fusion using Naïve Bayes as the basis of the Confusion Matrix Fusion.

Table 6: CM-Fusion Model's Normalized Confusion Matrix (Accuracy 0.93) for Vehicle 2 in Scenario 2:

	Precision	Recall	F-1 Score
Target Not Detected	0.90	0.81	0.86
Target Detected	0.94	0.97	0.96

Table 7: CNN Fusion Model's Normalized Confusion Matrix (Accuracy 0.85)for Vehicle 2 in Scenario 2:

	Precision	Recall	F-1 Score
Target Not Detected	0.64	0.84	0.72
Target Detected	0.94	0.85	0.89

Table 8: CM-CCA (Naïve Bayes Base) Fusion Model's Normalized Confusion Matrix (Accuracy 1.0)for Vehicle 2 in Scenario 2:

	Precision	Recall	F-1 Score
Target Not Detected	1.0	1.0	1.0
Target Detected	1.0	1.0	1.0

The results of Scenario 2 can be seen summarized below in Table 9. In the scenario, CM-CCA leads the fusion modalities in terms of performance, followed by the CM-CNN model. The base CNN in some cases can outperform the single source EO based models, but overall is unable to achieve a sufficient accuracy in comparison to the CM-CCA and CM-CNN models.

Table 9: Overview of single source and Fusion Model classification performance for targets in Scenario 2:

	Vehicle 1	Vehicle 2	Vehicle 3
EO Input	0.80	0.81	0.84
P-RF Input	0.62	0.63	0.62
CNN	0.78	0.85	0.81
CM-CNN	0.82	0.93	0.90
CM-CCA	**1.00**	**1.00**	**1.00**

For the CM-CCA models, the same four classifiers from the single source standalone models were applied, Naïve Bayes, Decision Tree, KNN, and Nearest Centroid. From the results of the four, the results remained unchanged for the CM-

CCA models, achieving the same F-1 score of 1.0 for each of the respective vehicles. The application of decision level fusion along with the use of the confusion matrix and CCA fusion provided the best results for classification.

6. CONCLUSIONS

The paper compares the decision-level fusion approaches on canonical variates analysis (CVA) and confusion matrix fusion (CFM) with deep learning methods (CNN) and traditional machine learning approaches for EO-RF data fusion for object classification. The DLF results demonstrate promise as shown on the ESCAPE data.

From parallel research with explainable AI using an optimized version of ProtoDash [42] and as clear from the results of the standalone P-RF results, the P-RF histograms are insufficient for classification on their own. From the initial results of that explainable AI research, the insights and impact of the P-RF data indicate that locally the P-RF CCA covariates provides a great impact, globally only second to that of the DOF-EO input. As such, in the future research we hope to implement other methods of using the P-RF data as in previous research that just the implementation of raw I/Q data was also insufficient. Additionally, exploring the impact of the CM-CCA fusion and comparing it with other models that are competitive in terms of performance is another area worth exploring.

Acknowledgements

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Air Force Research Laboratory, or the U.S. Government.

REFERENCES

[1] E. Blasch, E. Bosse, D. Lambert, *High-Level Information Fusion Management and Systems Design*, Artech House, MA, 2012.

[2] E. Blasch, A. Steinberg, S. Das, J. Llinas, C.-Y. Chong,*et al*., "Revisiting the JDL model for information Exploitation," *Int'l Conf. on Information Fusion*, 2013.

[3] L. Snidaro, J. Garcia, J. Llinas, E. Blasch, *Context-Enhanced Information Fusion*: *Boosting Real-World Performance with Domain Knowledge*], Springer, 2016.

[4] B. Kahler, E. Blasch, "Predicted radar/optical feature fusion gains for target identification," in *IEEE National Aerospace & Electronics Conference*, Fairborn, 2010.

[5] R. Falcon, R. Abielmona, et al., "Behavioral Learning of Vessel Types with Fuzzy-Rough Decision Trees," *Int'l. Conf. on Information Fusion*, 2014.

[6] C. R. Valenta, Z. J. Silva, "Towards single aperture RF/EO/IR systems: multi-spectral sensing and communication (Rising Researcher)," Proc. SPIE 11410, 2020.

[7] G. Taylor, R. Dehmubed, J. Cai, "Common EO and RF Antenna Apertures Using Integrated Device Combinations," in *IEEE Aerospace Conference*, 2003.

[8] G. L. DesAutels, B. M. Welsh, P. Beyerle, "Research and Development of an Integrated Electro-Optical and Radio Frequency Aperture," in *IEEE Aerospace Conference*, Big Sky, 2003.

[9] Z. J. Silva, C. R. Valenta, G. D. Durgin, "Optically Transparent Antennas: Survey of Transparent Microwave Conductor Performance and Applications," *IEEE Antennas and Propagation Magazine,* 2020.

[10] E. Blasch, R. Breton, and P. Valin, "Information Fusion Measures of Effectiveness (MOE) for Decision Support," *Proc. SPIE* 8050, 2011.

[11] P. Zulch, M. Distasio, T. Cushman, B. Wilson, B. Hart, E. Blasch," ESCAPE Data Collection for Multi-Modal Data Fusion Research," *IEEE/AIAA Aerospace*, March 2019.

[12] R Niu, P Zulch, M Distasio, *et al*., "Joint sparsity based heterogeneous data-level fusion for target detection and estimation," *Proc SPIE 10196*, 2017.

[13] P. Zulch, D. Shen, M. DiStasio, *et al*.**,** "Manifold Learning Algorithms for Sensor Fusion of Image and Radio-Frequency Data," *IEEE Aerospace Conf*., 2018.

[14] D. Shen, E. Blasch, P. Zulch, *et al*., "A Joint Manifold Leaning-Based Framework for Heterogeneous Upstream Data Fusion," *J. of Algorithms and Comp. Tech.* (JACT), 12(4):311-332, 2018.

[15] R. Niu, P. Zulch, M. Distasio, *et al*., "Joint Sparsity Based Heterogeneous Data-Level Fusion for Multi-Target Discovery," *IEEE Aerospace Conf*., 2018.

[16] D. Shen, J. Lu, P. Zulch, *et al*., "Joint Manifold Learning Based Distributed Sensor Fusion of Image and Radio-Frequency Data," *IEEE Aerospace Conf.*, March 2019.

[17] D Shen, P Zulch, M Disasio, *et al*., Methods, systems and media for joint manifold learning based heterogeneous sensor data fusion, US Patent App. 15/878,188, 2019.

[18] A. Vakil J. Liu, P. Zulch, E. Blasch, R. Ewing, J. Li, "Feature Level Sensor Fusion for Passive RF and EO Information Integration," *IEEE Aerospace Conference*, 2020.

[19] H. Zou, H. Lang., *et al*., "Covert Photo Classification by Deep Convolutional Neural Networks," *Machine Vision and Applications*, 28 (5-6): 623-634, Aug 2017.

[20] H. Zuo, H. Fan, *et al*., "Combining Convolutional and Recurrent Neural Networks for Human Skin Detection, "*IEEE Signal Processing Letters*, Jan 2017.

[21] U. Majumder, E. Blasch, D. Garren, *Deep Learning for Radar and Communications Automatic Target Recognition*, Artech House, 2020.

[22] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications* 23, 2031–2038 2013.

[23] G. Andrew, R. Arora, J. Bilmes, K. Livescu, "Deep Canonical Correlation Analysis," *ICML*, 2013.

[24] A-R. Mohammadi-Nejad, G-A. Hossein-Zadeh, H. Soltanian-Zadeh, "Multi-modal Data Fusion Using Group-structured Sparse Canonical Correlation Analysis: A Simulation Study," *Int'l Conf. on Pattern Recognition and Image Analysis*, 2017.

[25] H-T. Wang, J. Smallwood, J. Mourao-Miranda, *et al*., "Finding the needle in high-dimensional haystack: A tutorial on canonical correlation analysis," arXiv:1812.02598, 2018.

[26] Y. Zheng. E. Blasch, Z. Liu, *Multispectral Image Fusion and Colorization*, SPIE Press, 2018.

[27] Z. Liu, E. Blasch, V. John, "Statistical Comparison of Image Fusion Algorithms: recommendations," *Information Fusion*, 36, 251-260, 2017.

[28] B. Waske and S. van der Linden, "Classifying Multilevel Imagery From SAR and Optical Sensors by Decision Fusion," *IEEE Transactions on Geoscience and Remote Sensing*, 46, (5): 1457-1466, May 2008.

[29] G. Fasano, D. Accardo, A. E. Tirri, *et al*., "Radar/electro-optical data fusion for non-cooperative UAS sense and avoid," *Aerospace Science and Tech. 46*, 436–450, 2015.

[30] G. Fasano, D. Accardo, A. E. Tirri, A. Moccia, "Experimental analysis of onboard non-cooperative sense and avoid solutions on radar, optical sensors, and data fusion," *IEEE Aerospace and Electronic Systems Magazine*, 31(7), 2016.

[31] K. Yang, Y. Bar-Shalom, P. Willett, "Track-to-track fusion

with cross-covariances from radar and IR/EO sensor," *Int'l. Conf. on Information Fusion*, 2019.
[32] J. Liu, H. Mu, A. Vakil, *et al*., "Human Occupancy Detection via Passive Cognitive Radio," 20, 4248; *Sensors*, 2020.
[33] A. Vakil, J. Liu, P. Zulch, E. Blasch, R. Ewing, J. Li, "Feature Level Sensor Fusion for Passive RF and EO Information Integration," *IEEE/AIAA Aerospace Conf.*, 2020.
[34] B. Liu, E. Blasch, Y. Chen, D. Shen, G. Chen, "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier," *IEEE Intl Conf. on Big Data*, 2013.
[35] E. Blasch, *Derivation of a Belief Filter for Simultaneous High Range Resolution Radar Tracking and Identification,* Ph.D. Thesis, Wright State University, 1999.
[36] E. Blasch and L. Hong, "Data Association through Fusion of Target track and Identification Sets," *Int. Conf. on Info Fusion*, 2000.
[37] E. Blasch, K. B. Laskey, A-L. Jousselme, V. Dragos, P. C. G. Costa, and J. Dezert, "URREF Reliability versus Credibility in Information Fusion (STANAG 2511)," *Int'l Conf. on Info Fusion*, 2013.
[38] E. Blasch, S. Ravela, A. Aved, (eds.), *Handbook of Dynamic Data Driven Applications Systems*, Springer, 2018.
[39] J. Dezert, F. Smarandache, *Advances and applications of DSmT for information fusion (Collected works)*, Vols. 1-4, American Research Press, 2015.
[40] D. Harris, D. Dunham, "Addressing Assumptions About Expert Information Leveraging the Transferable Belief Model," *IEEE/AIAA aerospace*, 2010.
[41] F. Darema, E. Blasch, *et al*. (eds.), *Dynamic Data Driven Applications Systems*, International Conf. Springer, 2020.
[42] K. S. Gurumoorthy, A. Dhurandhar, G. Cecchi and C. Aggarwal, "Efficient Data Representation by Selecting Prototypes with Importance Weights," *Inter. Conf.on Data Mining,* pp. 260-269, 2019.

BIOGRAPHIES

Erik Blasch *is a program officer with the Air Force Research Laboratory. He received his B.S. in Mech. Eng. from the Massachusetts Institute of Technology and PhD in Electrical Eng. from Wright State University in addition to numerous Master's Degrees in Mech. Eng., Ind. Eng., Elect. Eng., Medicine, Military Studies, Economics, and Business. Additionally, his assignments include Colonel in the USAF reserves, adjunct associate professor, and president of professional societies. His areas of research include information-fusion performance evaluation, image fusion, and human-machine integration; compiling over 900 papers, 30 patents, and 7 books. He is an Associate Fellow of AIAA, Fellow of SPIE, and Fellow of IEEE.*

Asad Vakil *graduated the Ohio State University in 2018 with a degree in Electrical and Computer Engineering and is currently in the process of completing a master's degree in Electrical and Computer Engineering at Oakland University. His previous undergraduate internship experiences include product development for HVAC systems, UL validation, and power systems design over the span of six internships between Emerson Climate Technologies and the Eaton corporation. His current work as a graduate research assistant focuses on heterogeneous sensor fusion between passive radiofrequency and electro optical modalities for the purposes of achieving detection and tracking.*

Jai Li *received her B.S. degree in electronics and information systems from Peking University, Beijing, China, in 1996, the M.S.E. degree, and the Ph.D. degree both in electrical engineering from the Univ. of Michigan, Ann Arbor, MI, in 1997 and 2002, respectively. She has been a faculty member in the School of Engineering and Computer Science at Oakland Univ. since 2002. Her research interests are in the areas of statistical learning and signal processing with applications in radar, sensor fusion, communications and biomedical imaging. She has authored/co-authored over 70 referred publications, including one book. Her past and current research are sponsored by NSF, NIH, General Motors, Fiat Chrysler, National Research Council and AFOSR. Dr. Li serves as a member of technical committees of several international conferences and workshops. She is a senior member of IEEE.*

Robert Ewing *began his career in the Propulsion Laboratory at Wright-Patterson Air Force Base (Dayton, Ohio) with the development of high-speed jet engine control systems. He later held the position of a medical research scientist at the Aerospace Medical Research Laboratory and at the Air Force Institute of Technology (AFIT), where he co-pioneered the use of Quantitative Feedback Theory (QFT) for the design of robust analog and digital flight control and aircraft ejection systems. During the early 90's, Dr. Ewing led the Wright Laboratory's Solid State Electronic Devices Directorate's research in the area of mixed-signal hardware description language (VHDL-AMS) for microelectronic VLSI digital design, the IEEE VHDL standards and formal verification of digital designs. He has authored over 140 journal and conference publications, chaired numerous international IEEE conferences, Associate Editor for Electronic Letters, The Institute of Engineering and Technology (IET) Journal, and Board of Governors, Executive Director, IEEE Aerospace and Electronics Systems Society.*