Using Multimodal Input for Autonomous Decision Making for Unmanned Systems

"What it needs in order to evolve, is a human quality. Our capacity to leap beyond logic." – Capt. Kirk, Star Trek: The Motion Picture

James Neilan[1], Charles Cross[2], Paul Rothhaar[3], Loc Tran[4], Mark Motter[5], Garry Qualls[6], Anna Trujillo[7], and B. Danette Allen[8]
NASA Langley Research Center, Hampton, Virginia 23681

Autonomous decision making in the presence of uncertainly is a deeply studied problem space particularly in the area of autonomous systems operations for land, air, sea, and space vehicles. Various techniques ranging from single algorithm solutions to complex ensemble classifier systems have been utilized in a research context in solving mission critical flight decisions. Realized systems on actual autonomous hardware, however, is a difficult systems integration problem, constituting a majority of applied robotics development timelines. The ability to reliably and repeatedly classify objects during a vehicles mission execution is vital for the vehicle to mitigate both static and dynamic environmental concerns such that the mission may be completed successfully and have the vehicle operate and return safely. In this paper, the Autonomy Incubator proposes and discusses an ensemble learning and recognition system planned for our autonomous framework, AEON[23], in selected domains, which fuse decision criteria, using prior experience on both the individual classifier layer and the ensemble layer to mitigate environmental uncertainty during operation.

Nomenclature

LIDAR	=	Light Detection And Ranging
SONAR	=	Sound Navigation And Ranging
PM	=	Performance Matrix
S	=	Symbol Object or Weighted Sum. Defined in text.
A	=	User Input Symbol
B	=	Algorithm Decision Symbol
P( )	=	Probability Distribution
AEON	=	Autonomous Entity Operations Network
DDS	=	Data Distribution Service
Java	=	The Java Programming Language
C++	=	The C++ Programming Language
NASA	=	National Aeronautics and Space Administration
SLAM	=	Simultaneous Localization And Mapping

[1] Computer Engineer, NASA Langley Research Center, MS 492, Flight Software Systems Branch,
[2] Software Engineer, NASA Langley Research Center, MS 492, Crew Systems and Aviation Operations
[3] Research Engineer, NASA Langley Research Center, MS 492, Dynamic Systems and Controls Branch
[4] Computer Engineer, NASA Langley Research Center, MS 492, Flight Software Systems Branch
[5] Senior Research Engineer, NASA Langley Research Center, MS 492, Electronics Systems Branch, AIAA Member
[6] Senior Research Engineer, NASA Langley Research Center, MS 492, Aeronautics Systems Engineering Branch, AIAA Member
[7] Senior Research Engineer, NASA Langley Research Center, MS 492 Crew Systems and Aviation Operations, AIAA Member
[8] NASA Langley Autonomy Incubator Lead, NASA Langley Research Center, MS 492, Crew Systems and Aviation Operations, AIAA Senior Member

This work is a supporting component of the Autonomy Incubator initiative at NASA Langley Research Center in autonomous system implementation and management.

VO	=	Visual Odometry
INCA	=	"I've No Cute Acronym", Ensemble System
ELARS	=	Ensemble Learning And Recognition System
MAV	=	Micro Aerial Vehicle
GPS	=	Global Positioning System
NAS	=	National Air System
UAV	=	Unmanned Aerial Vehicle
AAV	=	Autonomous Aerial Vehicle
UxV	=	Unmanned (Land/Air/…) Vehicle

I. Introduction

The Autonomy Incubator at the NASA Langley Research Center in Hampton, Virginia, is an agile, start-up modelled initiative, tasked to invigorate autonomy research within NASA and push towards safe, reliable, and predictable autonomous solutions intended to participate both within and without the National Air System, NAS, framework. Multiple research initiatives are ongoing, covering topics like DDS based autonomy frameworks[23], controls and dynamics[24], reinforcement learning[25], go-around decision controllers[26], human computer interfacing and ground control systems[27], safe and robust operations[28], and test and evaluation[29]. The work described in this paper constitutes the autonomy and decision making component and how machine learning can be used for mission management of UxVs.

Machine learning and pattern classification methods are classically considered a subcategory of the Artificial Intelligence, or AI field of computer science. AI is defined in Tveter[2] as the field that studies the synthesis and analysis of computational agents that act intelligently. Computational agents are further defined as devices that act in a given environment whose decisions can be explained via a computational model[9].

The central scientific goal of the Autonomy Incubator research initiative is to understand the principles that make intelligent behavior possible on robotic platforms. This is accomplished by analyzing both natural and artificial systems, formulating and testing hypothesis, and designing, building, and evaluating computational systems that perform tasks whereby having intelligence is regarded as necessary[2].

The problem of using intelligence for recognition system robustness and reliability is an ongoing issue in machine learning. Tuning an algorithm to perform well is more of an art than a science[3]. Recognition algorithms often work "ok" but do not perform well when variances in lighting, symbol alignment, or scale occur in the input data. For instance, in Boas[12], tests were conducted at airports in Boston, Massachusetts, Dallas, Texas, and Fresno, California, where face recognition technologies were implemented in attempting to identify passengers. The results were considerably discouraging due to variations in lighting, skin textures, and scale and orientation components. Over and under fitting the recognition model to the training data also play roles in the poor performance of recognition systems. Bias and variance factors must be considered[9].

In attempting to decrease classification error and improve system robustness, researchers investigate three basic components of the classifiers and training sets. The first component, bias, measures the accuracy or quality of a classification match, while the second component, variance, measures the precision or specificity of the match. A high bias implies a poor match, and a high variance implies a weak match[3]. The third term, noise, is a measure of the amount of additional information that adversely affect the overall correctness and robustness of a recognition system

In statistical machine learning, we wish to model a function that transforms the input symbol to some output[11]. This function can be a regression problem, where the input symbol maps to a set of data values, like age, or height. The function can try to solve a category prediction problem where the input is mapped to a symbol and then executed as a command. With real world systems, noise invariably enters with the input and must be handled without knowing the amount or form of the noise data. Lighting variance, flickering of some background source not present in the training data, variations in training interpretation, improper correction factors, and many more are all considered sources of noise that can not necessarily be accounted for in the training set for a particular algorithm[9].

In fitting the model to the training data, we must consider the effects of bias and variance in order to improve performance with respect to symbol recognition. If our model is poor then we have a high bias problem where neither the training set nor test data perform well. However, if our method is too powerful, then we may experience a variance issue where our system performs well on the training data but poorly on the test data.

Researchers have proposed a general table for solving some of the major issues with bias, variance, and model construction. In Bradski[11], researchers from Willow Garage; a research company once focusing on robotics for consumer applications, created table 1 as a general guide to addressing these concerns.

Table 1 – Willow Garage's general solutions to machine learning problems.

Problem	Possible Solution
Bias	• More features can help make a better fit • Use a more powerful algorithm
Variance	• More training data can help smooth model • Fewer features can reduce overall fitting. • Use a less powerful algorithm
Good test/train, bad real world	• Collect a more realistic set of data
Model can't learn test or train	• Redesign features to better capture invariance in the data. • Collect new, more relevant data. • Use a more powerful algorithm.

Table 1 shows that bias and variance are tied in an inverse manner. To mitigate high bias, we select more discriminating features while using a more powerful algorithm, yet to mitigate high variance, the opposite is true. So a middle ground must be found such that we maximize classifier performance and minimize the negative effects of bias and variance[11]. This is often a difficult hurdle and illuminates the point that over-tuning an algorithm breeds internal deficiencies and thus limits the classifier to a highly defined task domain.

Two factors are of importance to a user in determining the performance of a recognition system. The first is the system's response time. For concerns like detection and avoidance the system is desired to behave with a high response rate. The second is the recognition algorithm's correctness. An algorithm is required to be consistently correct and reliable thereby maintaining a systems deterministic output.

It is difficult to achieve high recognition results using a single classifier due to many pattern variations which depend on deep prior knowledge not available in the training phase of standard recognition pipeline[13]. Overspecialization of an algorithm for maximal performance may cause it to become ineffective in environments in which it was not designed. Adjusting for the problems of bias and variance bounce algorithms between being too powerful or too weak. This problem has led researchers to using ensembles of weaker designed classifiers, combined correctly, that result in higher classification accuracy and robustness[9].

With these considerations in mind, ensemble systems are considered, consisting of weaker sets of classifiers, to better predict and track objects of interest. Klien[22] details the varied components of sensor and data fusion and how techniques like ensembles, particle filters, and other methods are used such that a system may learn, refine, and better handle dynamic conditions in a real world application domain. The Autonomy Incubator, using visual cameras, LIDAR, SONAR, thermal, and vehicle communication channels, are pushing towards a robust autonomy framework, AEON[23], Autonomous Entity Operations Network, aiding an autonomous entity’s decision making capability.

II. Background

A. Ensemble Methods

There are two distinct variations of ensemble systems in practice. The first system involves a single algorithm that is iterated over a training dataset to produce a number of classifiers. This is referred to as a dependent framework. The second system is one that combines a number of distinctly independent algorithms, each possessing a separate classifier. This is considered to be an independent framework[7].

With this understanding, an ensemble framework typically contains the following components[7]:

- Training set
- Base inducer
- Diversity Generator
- Combiner

The Training set is a labeled dataset used for the ensemble learning stage. This set contains a representation of the alphabet of symbols making up the decision space of possible classes.

The Base Inducer is the algorithm or set of algorithms that obtains a training set and forms the classifier(s). The classifier or set of classifiers represent the generalized relationship between the input features and target mapping to a class.

The Diversity Generator, for dependent ensemble approaches, generates the diverse classifiers over the alphabet of symbols in the decision space. In independent ensembles, this component is typically not used.

Finally, the Combiner brings together the classifiers and maps the sample space of classifier output to the decision space of the system whereby producing an overall recognition decision. Various ensemble decision and classifier combination methods exist.

Apart from the general framework and training set approach to the ensemble system, the combiner constitutes a significant component to a well-functioning system. Combination methods are varied and generally deal with the best vote or highest probability measure of a set of classifier decisions[9].

Standard combination techniques range from weighting methods, such as majority voting and classifier weighting, to Bayesian combinations methods and log opinion pools[7]. Whatever the approach taken, the general system must deal with three important issues.

- The response of the multiple classifier must be the best one given the results of each individual classifier.
- The possible responses may be of differing types and need to be combined in a coherent way.
- The ensemble must perform better than any individual in the system, otherwise there is no need for the ensemble[14].

Previous work in gesture and object detection and recognition systems using an ensemble of classifiers[1, 8, and 9] has shown substantial advantages to using ensemble systems as a means to better guess appropriate responses over an individual classifier approach. Starting in the 1970's, researchers have studied the benefits of a combined classifier approach such that the overall fused classifier system out perform an individual classifier in the ensemble[15]. Progress in ensemble systems gained greater interest in the 1990's with Hanson and Salman[4], suggesting that an ensemble of neural networks improved the predictive performance of a single network. Additional techniques, such as Bagging and Boosting as in Schapire[5] empirically demonstrated that a strong classifier can be constructed by incremental training of a series of weaker classifier algorithms by using the prior classifier performance as input into the next classifier in the set.

Bagging, boosting, and other techniques have been shown to improve approaches in remote sensing[16], incremental learning[9], and gesture recognition[1], highlighting vetted approaches to implementing ensemble solutions on production level systems.

In addition, there typically exists two standard ensemble methodologies. One method, generative; as with Adaboost[17], form the classifier ensemble by iterating over a classifier, modifying the training set on each pass over a given classifier to better support deficiencies in the overall recognition space. The series of generated classifiers are then fused such that each decision set gets passed along the series, with a final decision representing the best evidence-based supported selection. The second method, non-generative, do not actively generate classifiers but combine the independent classifiers after each component classifier has made a selection based on the problem space feature points. Figures 1 and 2 depict the two system architectures.

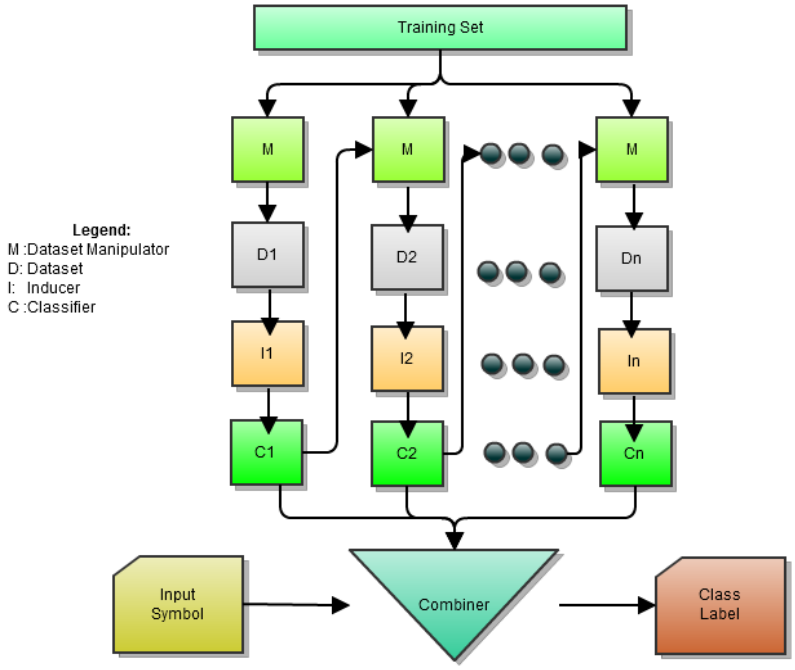

Figure 1 – Generative Ensemble System[9]

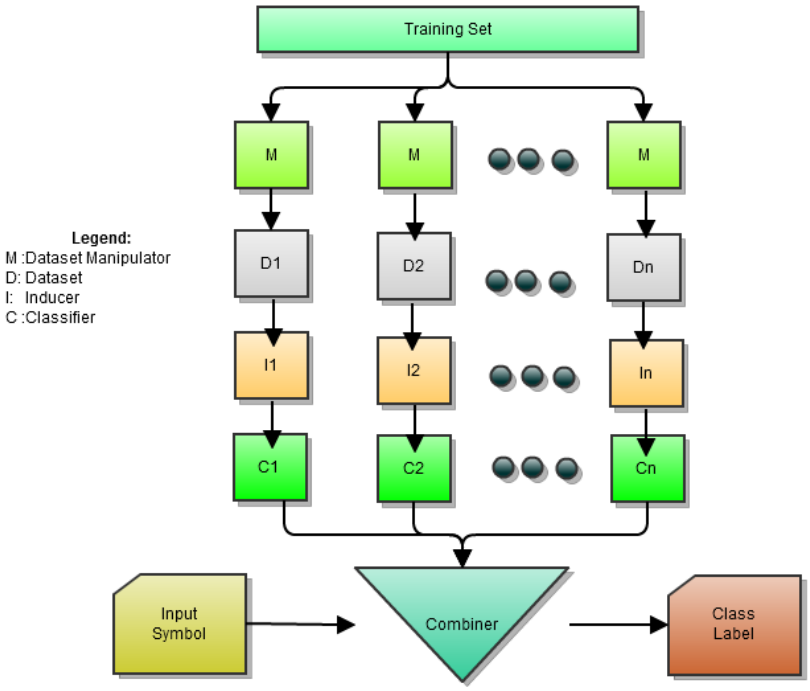

Figure 2 – Nongenerative Ensemble System[9]

It is important to consider the three types of classifier output typically found in these methods[1].

- Type 1: Abstract / Exact - Each classifier C_i outputs a single label given an input feature vector x pertaining to an unknown symbol *x*. Type 1 output contains no certainty measure as to the classifiers confidence in the mapping $C_i(x) = L_i$, with L_i representing a label in the classification space.
- Type 2: Rank – Each classifier output provides a ranked set of output values. The output is in a ranked order from most plausible to least and can be thought of as the statement "The symbol is most likely a 4, but could be a 3, and less likely a 2."This can be especially suitable for problems with a large number of classes[1].
- Type 3: Probability – Each classifier output gives the most information since either a ranking or classification can be produced from it[9]. Type 3 output is in the form of a probability distribution of the unknown input symbol over the recognition alphabet. Here, $C_i(x) = \{a_1(x), \ldots, a_L(x)\}$, is the set of probabilities that classifier C_i considers the unknown input symbol *x* as belonging to class $a_1 \ldots a_L$.

From Newell, Neilan and Henderson[1], both Parker[10] and Sannen[7] state that type 1 classifiers output provide the least benefit to classifier fusion since alternative solutions are not given. Type 2 classifiers output offer relative information about possible symbol class labels, giving a rank comparison between each symbol. This type does not give absolute information regarding how the classifier performed over the entire sample space. Type 3 classifiers output offer the most information about alternative class possibilities.

We focus on a non-generative approach in this paper due to three main reasons supported by the methodology:

1. Modularity in Input Modality
2. Modularity in Implementation Domain
3. Modularity in Plug-n-Play Algorithm Selection

Our software framework implements the DDS (Data Distribution Service) protocol which allows us to easily and reliably modularize our autonomy components, fostering plug-n-play component capability, among other advantages discussed in greater detail in Cross et al.[23]. With this framework, AEON (Autonomous Entity Operations Network), we are able to develop and implement both novel and COTS (Consumer Off The Shelf) solutions quickly and with immediate quantitative feedback. ELARS, Ensemble Learning And Recogition System, is currently being implemented in AEON, and we intend to determine application effectiveness in an indoor, cluttered flight pattern, using quad-rotor air frames.

The proposed system, ELARS, uses Bayesian inference, described below, and is based off of work done by Newell, Neilan, and Henderson[1].

B. INCA[1]

Newell, Neilan, and Henderson[1] describe INCA ("I've No Cute Acronym"), as a non-generative Bayesian based meta-algorithmic model that combines individual recognition results of the component algorithms to produce an overall recognition decision which is no worse than the individual algorithmic output. Empirically, the model has proven to produce results 10~20% above the component methods in the gesture recognition domain of hand-sign and hand-written symbols.

As described in Newell, Neilan and Henderson[1], the INCA model assumes the existence of *N* different recognition algorithms. Each of the algorithms contain inherent strengths and weaknesses and tend only to be strong in subsets of the application domain, while weak over the remaining recognition set contained in the domain. The key to INCA's performance was the effectiveness in the use of inherent algorithmic weaknesses and consistency of those weaknesses. For example, if an algorithm misrecognizes a handwritten '*H'*, as a '*K'*, then this information is useful and allows a researcher to modify the importance of the algorithm's result. Figure 3 gives an overview of the INCA data flow.

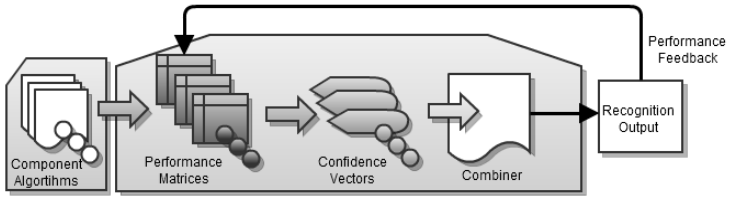

Figure 3 – INCA layout[1]

Bayes' Theorem is based upon the concept of conditional probability. Bayes theorem is presented below. It applies conditional probability to a partition of some sample space of mutually exclusive and exhaustive sets $(A_1,A_2,A_3 \ldots A_m)$. The theorem supplies a formula for $P(A_i \mid B_k)$ where B_k is some empirically observable event. For our purposes, B_k is the recognition result produced by some algorithm and the A_i values represent the event of some symbol having actually been entered as input[1, 8].

$$P(A_i|B_k) = \frac{P(B_k|A_i)P(A_i)}{P(_k)} \tag{1}$$

The power of INCA comes from the use of Bayes' theorem, or more explicitly, the confusion matrix built of the performance metrics over the recognition domain to support future recognition decisions. A performance matrix for any component algorithm is an $M \times M$ matrix where M is the number of symbols in the recognition domain. For any recognition algorithm A_i the performance matrix *PMi [x][y]* contains the number of times that, during previous recognitions, the algorithm A_i recognized the actual input gesture S_x as the gesture S_y. After applying Bay's theorem to the data stored in the confusion, or performance matrix, INCA can determine the probability that any gesture was

given, here denoted as S_x, was in fact the actual gesture entered given the algorithms recognition of S_y as its result. This corresponds to the conditional probability $P(S_x|S_y)$, and given as:

$$P(A_i|B_k) = \frac{PM[S_i, S_k]}{\sum_{j=1}^{N} PM[S_j, S_k]} \tag{2}$$

As an example, figure 4 depicts the confusion matrix for a sample algorithm and its confidence vector for the recognition of the digit '1'.

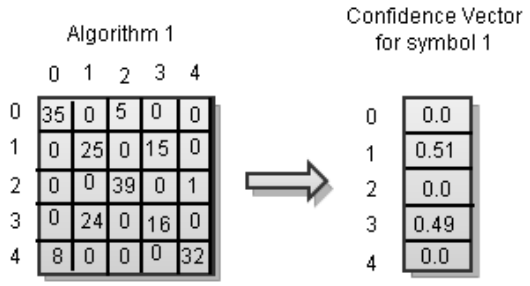

Figure 4 – Confidence vector[1] representing the probability space for the algorithm recognizing the symbol "1" and the INCA probability that the user did in fact enter a "1".

Here, it is shown that the entry $PM_i\ [x][y]$ in the matrix over the sum of the column $PM_i[*][y]$. Once this calculation is carried out for each gesture in the given alphabet we can build its confidence vector. We can then proceed to combine it with the vectors of all other algorithms to obtain the overall system recognition result.

Considering the three types of classifiers in a non-generative system, Newell et al.[1] has identified that the INCA model from previous work converts type 1 and 2 information from individual classifiers into type 3 via the performance metrics tracking and analysis.

III. ELARS- Ensemble Learning And Recognition System

In Neilan[9], INCA was extended into the American Sign Language, ASL, hand-signed and handwritten character recognition domain, testing three fusion methods of the component algorithm confidence vectors and comparing against a more memory intensive method known as Behavior Knowledge Spaces, BKS. The system, ELARS, Ensemble Learning And Recognition System, was a Java based application that accepted two input modalities; a pen tablet and data glove, and compare fusion techniques in the gesture recognition domain.

In the practice of data fusion, INCA represented the first step in the standard fusion pipeline, lending itself to testing fusion methods to better support a systems autonomous decision given supporting or conflicting evidence.

ELARS extended the INCA concept by implementing machine learning techniques coupled with a pen tablet and data glove modalities. Figure 5 defines the system layout.

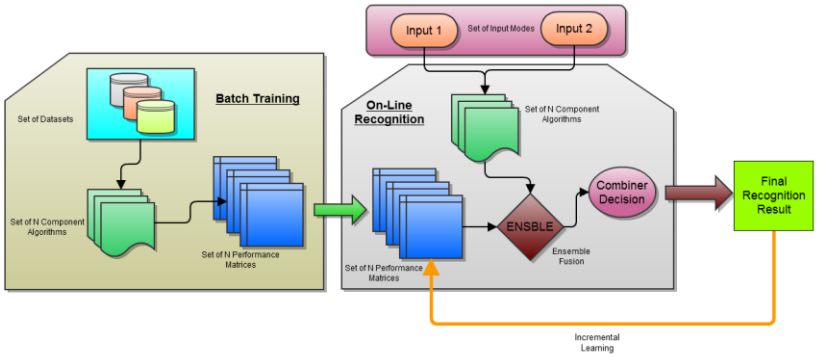

Figure 5 – ELAR System Layout[9]

The component algorithms in ELARS are disjoint and independent of one other. The system is initially trained in a batch mode configuration, accepting a training set or sets for each algorithm. Once the training is complete, the performance matrices are saved to an ".xml" file. These files are then used to initialize the performance matrices for the on-line component of the system.

The on-line configuration represents the entity recognition engine. Symbol input is received via the input layer consisting of the desired input device(s). The symbol is passed to the component algorithms and each algorithm produces a mapping to known label. The ensemble component then takes the input symbol guesses from the component algorithms and the performance matrices corresponding to each algorithm to create a set of corresponding confidence vectors. The confidence vectors are then combined and a final decision is given[9].

The autonomy implementation of ELARS consists of the learning feedback branch for the overall system, allowing for not only the individual classifier priors to play a role in the decision process, but the overall system performance given environmental considerations are used to better support a decision. Extending this into a multi-modal input functionality, ELARS will listen to visual camera information, LIDAR, SONAR, and communications information to better handle the flight environment and mitigate risk while executing a desired mission.

We briefly discuss single and multiple input modality ensemble systems in the community as examples of how ELARS may be used in a domain agnostic manner.

C. Single Input Modality Classifiers

In the domain of handwritten character recognition, Xu, Krzyzak, and Suen present results of four expert algorithms, each involving techniques of skeleton and contour feature extraction for recognizing handwritten digits[18]. The database used was from U.S. Zipcode database of the Concordia OCR research team and consisted of 6000 samples with 400 samples for 10 numeral digits. 4000 of the samples were subsequently used for expert training, with 2000 symbols saved for validation.

The experts, or algorithms, were first tested individually, performing with 86.05%, 93.10%, 92.95%, and 93.90% accuracy for experts "1" through "4" respectively. The four experts where then combined using Dempster-Shafer (DS), naïve Bayes(NB), and voting methods described in section 2.2. Xu et al. confirmed that each combination method did in fact improve overall recognition over the individual methods in the ensemble.

The Dempster-Shafer method performed the best with an overall recognition of all digits with 98.95% accuracy with a rejection rate, or rate in which the input symbol was not classified, of 0.02%. Xu et al. also defined system reliability as with DS giving a reliability measure of 99.15% overall[9].

The NB approach had slightly lower accuracy but higher reliability with an overall recognition rate of 95.0%, a rejection of 5.0%, and a reliability measure of 100%. The voting method they used subsequently performed with a recognition accuracy of 94.3%, a rejection of 5.7%, and a reliability measure of 100%.

Xu et al. go on to summarize their work with some interesting observations. First, noting if the confusion matrices of the individual experts are well learned, then the NB formalism is typically the best combination method. The DS approach is robust in general and inaccurate learning does not influence the performance substantially[18]. Both DS and voting methods behaved well overall and DS was a better method if high reliability is required. In conclusion, Xu, Krzyzak, and Suen claim that the DS method is best overall, but other methods also provide performance enhancement in the domain of unconstrained digit recognition.

Sannen, Lughofer, and Van Brussels describe an incrementally adapting or learning ensemble framework used in five real-world quality inspection tasks from an industrial CD imprint production process and five data sets from a data repository[7]. The framework uses an incremental clustering method as an ensemble combination strategy, with Naïve Bayes, eVQ, and K-Nearest Neighbor approaches for the independent recognition algorithms.

D. Multiple Input Modality Classifiers

Schwenker, Scherer, Schmidt, Schels, and Glodek propose a system combining different input modalities for improving HCI via a system that recognizes human emotional state[19]. Schwenker et al state that "...human emotions are expressed through different modalities such as speech, facial expressions, hand or body gestures..." and coupled with the fact that research in affective computing is in uni-modal recognition systems; a multi-modal approach would be more accurate and robust against outliers caused by noise or miss-recognitions. Schwenker et al propose a prototype system using an audio-visual laughter detection and facial expression system for human emotion recognition[19].

Schwenker et al further investigated the facial detection system, building an algorithm level ensemble, using orientation histograms, principle components, and optical flow methods[19]. Each method modeled regions for recognition covering the face, mouth, right eye, and left eye. The twelve models where then fused using a voting method and a probability fusion technique.

The voting method performed with an 81.7% accuracy, improving on the 57.4% average rate of the individual component models. The probabilistic method performed slightly better than the voting method with an accuracy of 85% over the individual models' performances[19].

The laughter detection component utilized a Recurrent Neural Network, RNN, in order to detect laughter in natural conversations, consisting of a sparsely connected 1500 node network[19]. The RNN approach was then selected to provide recognition for both the audio and visual systems in the multi-modal system, however, no data was given to support this decision.

The audio and video data sets for the combined system was built using conversations between four people sitting around a table, recorded with a 360 degree camera and a centrally placed microphone[19]. The two separate RNNs, one for each modality, were used to detect laughter in the audio and video data, using a probabilistic fusion approach as the ensemble combiner. Individually, the audio and video, using the RNNs, performed with 87% and 82% accuracy, respectively. The system, combined, was able to detect laughter with a 91% accuracy over the two independent approaches.

Oza and Tumer[20] give a review of ensembles in real world applications. One interesting example is that of person recognition using an ensemble of independent classifiers from different input modalities. Oza and Tumer state that person recognition is historically one of the most frequent application domains for ensemble learning systems, and that combining diverse features into one recognizer is difficult because of scaling between the input methods. For instance, iris detection and classification differs from voice, which also differs from approaches applied to face recognition. "Ensembles consisting of individual recognizers for each modality would work better because they combine at the decision level where the scales would be the same."[20].

A multi-modal person recognition system is referenced by Oza[20] and described in Erdogen et al.[21]. The application domain was vehicle driver recognition, stating the benefits of the application as:

- Ensuring that only authorized drivers drive the vehicle.
- Personalizing the vehicle for the driver's physical and behavioral characteristics.
- Warning the driver and appropriate authorities if the driver is not in the proper condition to drive.
- Allowing for secure transactions, such as banking, from within the car.

The independent features that are pulled for the recognition task are face, speech, and behavioral characteristics such as pedal and steering input. Erdogen et al. use these 3 input modalities, combining the output from each modality using a weighted score summation method.

$$S = \sum_{k=1}^{N} w_k S'_k \tag{3}$$

Where S is the weighted sum of new scores for each validation test case, w_k as fixed weights and S_k' is a sigmoid normalization function which maps the scores from 0 to 1:

$$S'_k = \frac{1}{1 + e^{((-S_k - \mu)/\sigma)}} \tag{4}$$

With μ and σ being the mean and standard deviation of old scores obtained from their validation set.

Erdogen et al. report that separately, the recognition techniques of audio, face, and driving gave 98%, 89%, and 88.25 % accuracy, with combination of audio and driving, face and driving, audio and face, and all three combinations accuracy of 99%, 98%, 99.75%, and 100% respectively[9].

E. ELARS Performance – Gesture Recognition

Neilan[9] presents overall data on the performance of a three classifier esemble system for gesture recognition using a pen tablet and data glove to recognize hand signed and hand written digits 0-9. Tables 2 and 3 represent the three different fusion techniques and there comparison to the BKS method of ensemble analysis.

Table 2 – Data Glove[9]

%	MaxSum	DecisionTemplate	Dempster-Shafer	BKS
Accuracy	97.8	83.1	85.75	98.3

Table 3- Pen Tablet[9]

%	MaxSum	DecisionTemplate	Dempster-Shafer	BKS
Accuracy	62.2	7.8	34.4	62.4

As stated in Newell[1], the INCA model and, by extension, ELARS perform best in a user-dependent manner when considering the pen tablet modality. The data glove performed well overall and provided an interesting modality for the small test sets were used. Performance analysis proves difficult when considering the many characteristics of the recognition algorithm and ensemble approaches, however, we did find that as intricate a fusion approach may be, as with Decision Templates and Demspter-Shafer, does not mean that they can handle inconstant recognition of the component classifiers in an elegant manner. We found that the MaxSum approach to label selection was the most accurate method using the INCA model ensemble. The BKS method performed better than the MaxSum by 1.4% only in the 3-algorithm test using the data glove input modality alone[9].

F. ELARS for Autonomous Decision Making

The ELARS framework has been constructed both in Java and C++ to allow for ease in development for researchers within the Autonomy Incubator at NASA Langley. The ELARS Java and C++ library contains the following classes:

- Alphabet
- Symbol
- Ensemble
- Algorithm
- Performance Matrix
- Confidence Vector

At the bottom level, the Symbol class represents an object in the recognition space, e.g. a ‘car’, ‘tree’, ‘mav’. The collection of Symbols makes up the recognition alphabet encapsulated in the Alphabet class. Above the Alphabet lies the Algorithm level that encapsulates the independent classifier methods that are intended to make up the Ensemble. The Performance Matrix class captures the overall performance of the individual classifiers and the ensemble system. The Confidence Vector subsequently uses the individual performance matrices to build the probability space of the ensemble and fuses the vectors given a maximum summation or a more involved fusion method like Dempster-Schafer to best estimate the classification result. Figure 6 depicts the Java class hierarchy.

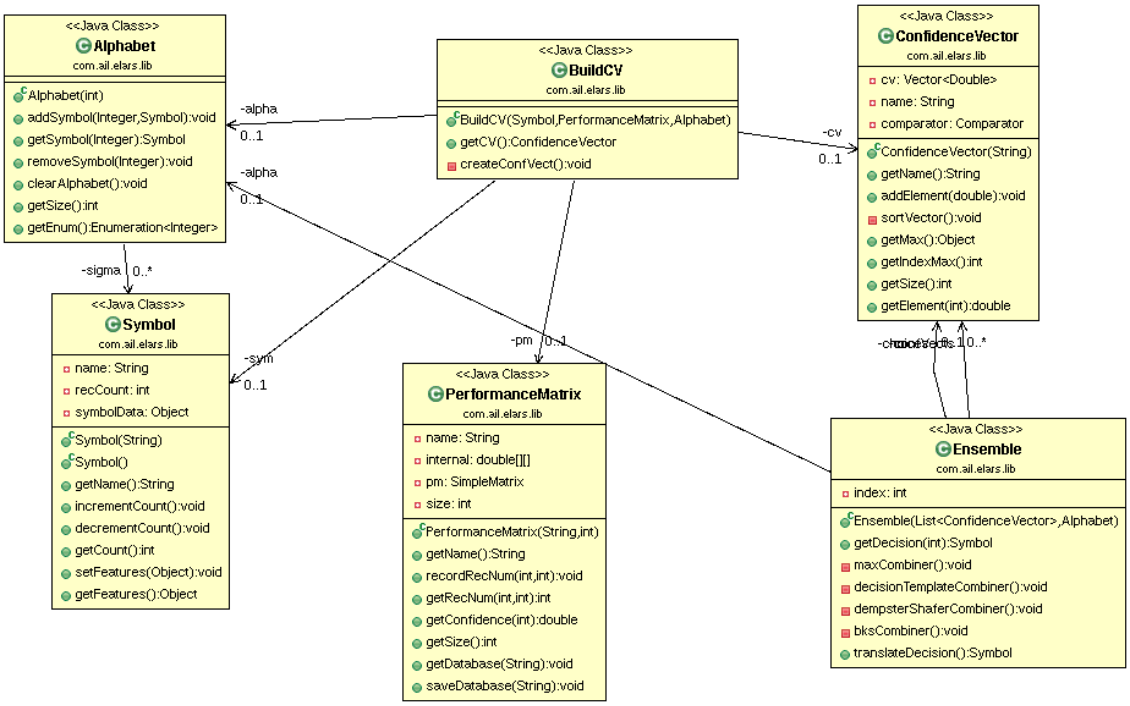

Figure 6 – ELARS Java UML

How can ELARS help flight autonomy on a MAV? We are considering the various application domains for autonomous flight in the Autonomy Incubator indoor flight facility and have determined that ELARS can assist in these main errors:

- System Health Assesment
- Object Recognition
- Collision Avoidance
- Person Detection
- Static/Dynamic Obstacle Identification

Our first application domain is that of Object recognition in building a flight ontology within our local airspace.

IV. Discussion

It can be argued that multi-modal input may possibly negatively impact recognition tasks given highly conflicting recognition guesses from each mode. E.g vision saying an object is a 'tree' and the LIDAR reports a 'moving car'. However, given the power of Bayesian inference and the ability of ELARS to correctly guess an object even in conflicting circumstances, supports our decision to continuing investigating the approach and measure it against state-of-the-art systems flying today.

G. Multi-Modalities on MAVs

What are some input modalities that make sense for a micro aerial vehicle, such that combining the modalities gives the MAV an advantage over more traditional, single sensor modality platforms? Though research is ongoing, in regards to state estimation and target tracking, it is clear that a mix between internal system, e.g. IMU, visual odometry, and external systems, e.g. GPS, Cell tower triangulation, ultra wide band radar, vision (monocular and stereo), LIDAR, and SONAR seem best to estimate the vehicles state, and the state of objects surrounding it.

The Autonomy Incubator is currently flying quad-rotor systems with IMU, visual odometry, monocular vision, and multi-agent communications as input modalities into an agents decision tree for mission execution. We are also interested in how failures of a given modality can be mitigated by another sensor systems and/or swapping performance priors given modifications to the environment during mission executions. We envision that rain may negatively affect RADAR signals, thus another modality, such as LASER or RGB vision may better perform for a given mission space. Also, there could be instances of performance priors swap if we know how a sensor operates in a certain adverse environment, in our case a rain storm, and subsequently adjust the responses for that sensor.

H. A Path to Autonomy

As a decision framework, ELARS is a Bayesian inference model, amenable to domain transition and is recognition alphabet agnostic, i.e. ELARS places no constraints on application domain nor object recognition classes. Adaptation onto a flight vehicle as exists in the Autonomy Incubator at NASA Langley encompasses a stepwise procedure, starting with indoor flight operations within a constrained configuration space.

Since the focus of autonomous flight at the Incubator consists of GPS-degraded/denied flight operations, the first implementation domain is naturally vision based detection and object recognition. We intend to use ELARS for both static and dynamic object identification, leading to better estimates on state, both external and internal, which feeds into the decision tree, allowing for the vehicle to better build a course of action, even if that action means flight termination.

V. Conclusion

There exists no lack of research effort in regards to detection and avoidance in all fields of UxV research. Many approaches are being pursued, allowing for fast image segmentation, detection and classification, and learning from visual markers. What is interesting about multimodality in classification tasks is that sensor modalities can complement each other and adjust performance characteristics given on-board sensor health and environmental concerns. E.g. heavy rain occluding RADAR data, and so forth. Other sensors can assist and varying performance matrices for certain known conditions can be swapped in real time to better support a mission task or tasks. The Autonomy Incubator is looking into ensemble learning in order to discover these modality characteristics, support mission dynamics, and enhance classification of obstacles in the sensing range of the vehicle.

The described Bayesian inference approach can be used to construct an initial set of known objects (training), distinguish between similar yet different objects in real-time, and update the systems belief space with respect to the world map[29]. This classification is possible even in the presence of conflicting information from independent methods or algorithms contributing to the system wide sensing and perception capability[29].

The Autonomy Incubator is pushing forward in this and other initiatives in order to develop an autonomous capability that is state-of-the-art and top-of-class, worldwide.

Acknowledgments

This paper describes the ongoing effort of all related members of the Autonomy Incubator at NASA Langley and constitutes multiple disciplines. All work could not be possible without the drive, energy, and dedication of the AI members and families. Student efforts during the spring of 2015 have also aided this work. Special thanks goes to Gil Montague, Matt Mahlin, Irvin Cardenas, Jacob Beck, and Sarah Voorhies. An additional acknowledgement goes to Dr. Gary Newell, Associate Professor of Computer Science, Northern Kentucky University, for his development of INCA[1] and guidance during the original ELARS work[9].

References

Periodicals

[1]G. Newell, J. Neilan, M. Henderson, "Probabilistic Gesture Recognition", Accepted to The *International Conference on Artificial Intelligence, Las Vegas Nevada. 2012.* n.p. 2012.

[2]D. R. Tveter, *The Pattern Recognition Basis of Artificial Intelligence,* IEEE Computer Society, 1998. ISBN 0-8186-7796-1

[3]R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2001.

[4]L. Hansen and P. Salmon, "Neural network ensembles.", *IEEE Transactions on Analytical Machine Intelligence*. Vol 12, pp. 993-1001.1990

[5]R. Schapire, "The Strength of Weak Learning", *Machine Learning*, vol 5, pp. 197-227, 1990.

[6]N. Oza and K. Tumer, "Classifier ensembles: Select real-world applications." 2007 Elsevier B.V. 2007.

[7]D. Sannen, E. Lughofer, and H. Van Brussel, "Towards incremental classifier fusion," *Intelligent Data Analysis*, vol. 14, no. 1, pp. 3–30, 2010.

[8]G. Newell, "A Probabilistic Approach to Gestural Recognition and Dialogue Management", *PhD Dissertation, unpublished*. UMI Bell and Howell publishing. The University of Arizona, 1995.

[9]Neilan, J., "Gesture Recognition Using Ensembles of Classifiers", M.S. Thesis, Northern Kentucky University, 2012, 122 pages. UMI Dissertations Publishing.

[10]J. R. Parker, Algorithms for Image Processing and Computer Vision, 2nd edition, Wiley Publishing, 2011.

[11]G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Publishing, 2008.

[12]G. Boas, "Face Time", *Photonics Spectra*, n.p. August

[13]F. Roli, G. Giancinto, G. Vernazza, "Methods for Designing Multiple Classifier Systems", *LNCS Multiple Classisfier Systems*, Springer Berlin vol 2096, pp. 78-87. 2001.

[14]T. Windeat and G. Ardeshir, "Decision Tree Simplification for Classifier Ensembles." International Journal of Pattern Recognition and Artificial Intelligence Vol. 18, No. 5 (2004) 749 World Scientific Publishing Company, 2004.

[15]Tuduran, C., Naegoe, V., "A NewNeural Network Approach for Visual Autonomous Orad Folloing", Latest Trends on Computers, vol1. 2010.

[16]L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1_2, pp. 1-39, Nov. 2009.

[17]P. Garg, N. Aggarwal, and S. Sofat, "Vision Based Hand Gesture Recognition." World Academy of Science, Engineering and Technology Vol. 49 2009.

[18]L. Xu, A. Krzyzak, C. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22, No. 3, May/June 1992.

[19]F. Schwenker,, S. Scherer,, M. Schmidt,, M. Schels,, and M. Glodek, "Multiple Classifier Systems for the Recogonition of Human Emotions." N. El Gayar, J. Kittler, and F. Roli (Eds.): MCS 2010, LNCS 5997, pp. 315–324, 2010. Springer-Verlag Berlin Heidelberg 2010, 2010.

[20]N. Oza and K. Tumer, "Classifier ensembles: Select real-world applications." 2007 Elsevier B.V. 2007.

[21]H. Erdogen, A. Ercil, H.K.Ekenel, S. Y. Bilgin, I. Eden, M. Kirisci, H. Abut, "Multi-modal person recognition for vehicular applications", *Proceedings of the sixth annual International Workshop on Multiple Classifier Systems*, Springer, Berlin, 2005. pp.366-375.

[22]Klein, L., "Sensor and Data Fusion: A Tool for Information Assesment and Decision Making", Second edition, SPIE Press, Bellingham, Washington USA.2012

[23]Cross, C., Motter, M., Neilan, J., Tran, L Qualls, G., Rothhaar, P., Trujillo, A., Allen, D., "it's Difficulte to work in groups when you're Omnipotent, So don't be – An Open, Distributed Software Architecture for UXS Operations", *Aviation 2015*, Dallas, TX.

[24]Rothhaar, P., Cross, C., Tran, L Motter, M., Neilan, J., Qualls, G.,., Trujillo, A., Allen, D., "A Flexible Flight Control System for Rapid GNC and Distributed Control Development", *Aviation 2015*, Dallas, TX.

[25]Tran, L., Cross, C., Motter, M., Neilan, J., Qualls, G., Rothhaar, P., Trujillo, A., Allen, D., "Reinforcement Learning with Autonomous Small Aerial Vehicles in Cluttered Environments", *Aviation 2015*, Dallas, TX.

[26]Motter, M., Neilan, J., Cross, C., Tran, L., Qualls, G., Rothhaar, P., Trujillo, A., Allen, D., "His Pattern Indicates Two-Diensional Thinking – Deciding to Go Around via Machine Learning", *Aviation 2015*, Dallas, TX.

27Trujillo, A., Cross, C., Tran, L Motter, M., Neilan, J., Qualls, G., Rothhaar, P., Allen, D., "I'm a Doctor, Jim Not and Engineer – Collaborating with Autonomous Agents", *Aviation 2015*, Dallas, TX.

[28]Qualls, G., Cross, C., Tran, L Motter, M., Neilan, J., Rothhaar, P., Trujillo, A., Allen, D., "Operating in Strange new worlds an dmeasureing Success", *Aviation 2015*, Dallas, TX.

[29]Allen, D., Cross, C., Tran, L Motter, M., Neilan, J., Qualls, G., Rothhaar, P., Trujillo, Crisp, V., "Towards Safe Robust Autonomous Operations", *Aviation 2015*, Dallas, TX.