

# Multi-Class Classification of Vegetation in Natural Environments Using an Unmanned Aerial System

Alistair Reid

Fabio Ramos

Salah Sukkarieh

**Abstract**—This paper presents an automated approach for the classification of vegetation in natural environments based on high resolution aerial imagery acquired by a low flying Unmanned Aerial Vehicle (UAV). Standard colour and texture descriptors are extracted on a frame by frame basis to build a representation of appearance, which is probabilistically classified by a novel multi-class generalisation of the Gaussian Process (GP) developed for this work. A GP approach was selected for probabilistic outputs, and the ability to automatically determine the relevance of each input dimension to each of the  $C$  classes in the problem. When learning hyperparameters from  $N$  training examples, the new formulation scales at  $\mathcal{O}(N^3)$ , rather than  $\mathcal{O}(CN^3)$  for the standard one-vs-all approach. The novel classification framework is trained and validated on a set of manual labels, and then queried to visualise a map of vegetation type under the UAV flight path. Mapping results are presented for a region of farmland in Northern Queensland, Australia that is infested with two invasive introduced tree species.

## I. INTRODUCTION

An approach has been developed for the automated classification of vegetation species in a natural environment, and applied to the mapping of invasive woody weed species over farmland in Northern Australia. Invasive plant species cause significant damage to the productivity of agricultural land every year, motivating investment in new detection technologies [1]. To provide the necessary information to support targeted eradication or containment strategies, we are developing an automated solution based on aerial imaging with an unmanned aircraft, followed by automated vision-based classification and mapping. The aim of this paper is to present a generic classification approach based on machine learning that is well suited to the high resolution colour imagery acquired by the UAV.

The UAV used to collect the dataset for this work was designed as a robust, low cost sensor platform to operate from remote survey sites. The airframe is derived from a one third scale J3 Cub model, and has been outfitted for autonomous flight with a CloudCap Piccolo flight controller, a suite of navigation sensors including an inertial measurement unit, a GPS unit, and a downward pointing monocular video camera. The aircraft and payload are shown in Figure 1.

This work is supported in part by Meat and Livestock Australia (MLA) under project code B.NBP.0474, UAV Surveillance Systems for the Management of Woody Weeds. This work is also supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors also acknowledge Mitch Bryson for providing the UAV navigation framework to support the map reconstruction results.

A. Reid, F. Ramos and S. Sukkarieh are with the Australian Centre for Field Robotics, The University of Sydney, NSW, 2006, Australia {a.reid, f.ramos, s.sukkarieh}@acfr.usyd.edu.au



Fig. 1. The autonomous UAV, based on a J3 Cub 1/3 scale model. Sensor payload includes IMU, GPS receiver, downward pointing monocular camera and a PC104 stack.

For the results presented in this paper, four classes of tree have been defined: *Prickly Acacia* (invasive), *Parkinsonia* (invasive), *Eucalypt* (the only native trees surviving in the area) and *null* features that were not of interest. In identifying these features, the colour imagery acquired by the UAV offers less spectral discrimination power than typical hyperspectral satellite imagery, but because the UAV is able to fly at low altitudes (100m above ground) it can provide a very high spatial resolution of 4cm/pixel that resolves unmixed colour information, and enables the use of *image texture* to describe shading and pattern. A supervised machine learning approach based on Gaussian Processes (GPs) has been used to map generic appearance descriptors into class probabilities.

GP classification is a state-of-the-art method for handling difficult classification problems [2]. The Bayesian formulation infers probabilistic outputs, and provides a closed form marginal likelihood that can be optimised to select an optimal model[3]. GP learning with this criterion is resilient to overfitting, and can automatically determine the relevance of the feature dimensions to the classification problem [4].

In a  $C$  class problem with  $N$  training labels, a pre-trained GP model is relatively fast to query at  $\mathcal{O}(NC)$  and therefore well suited to processing the large image datasets in this work. However, model selection can be slow, as the cost of training a GP is dominated by the inversion of a (potentially) large covariance matrix, and the learning criterion may need to be evaluated many times when optimising within a high dimensional hyperparameter space. The usual approach to a multi class problem is to define  $C$  one-vs-all binary models, leading to a cost of  $\mathcal{O}(CN^3)$  to process the examples or evaluate the learning criterion. In this work, an alternative generalisation is proposed that places a common covariance model over the  $N$  training labels that can be solved at an improved  $\mathcal{O}(N^3)$ . This new framework is cross validated on training data to obtain performance statistics, and then

applied open-loop to the UAV imagery to construct a tree species map over a region of interest. The reader is directed to [5] for details of the navigation data fusion.

The remainder of this paper is organised as follows. Section II reviews similar and related work. In Section III, the classification framework and essential GP background theory is described. Finally, the experimental setup, and the results including maps of the region of interest are presented in Section IV.

## II. RELATED WORK

### A. Aerial Classification

Vision and perception in aerial data has been studied in a broad range of applications, both in the robotics community and outside. In the robotics community there has been recent interest in aerial classification of terrain types such as roads, trees, grass and buildings for assisting a ground vehicle using sensors such as colour cameras and/or LIDAR (Light Detection And Ranging), although these approaches have not been concerned with vegetation type [6], [7].

Mapping of vegetation type over large natural environments is necessary for problems such as geostatistical studies of land cover, or the detection and monitoring of invasive species[1]. Large scale ecological mapping is usually approached using multi-spectral imagery from high flying aircraft [8], commercial satellites [9], or multi-temporal public datasets such as from the Landsat Thematic Mapper satellite [1]. Unlike our vision perception based approach, these studies have relied primarily on pixel spectra: peaks in the visible and near infrared bands are associated with cellular chemical and biological properties of the vegetation [8]. Examples of vegetation type classification using colour imagery have previously been restricted to controlled agricultural environments such as crop fields, orchards or plantations [10], where visual properties such as co-occurrence matrix statistics [8], or Gabor Filter banks [11], have been used to compensate for a loss of spectral resolution. The gap between automated robotics and the study of natural environments is being bridged, as we are now starting to see autonomous UAVs like ours appear in agriculture and ecological mapping applications [12], [13]. The primary role of these UAV platforms is to provide current imagery over large regions of interest, and to obtain a sufficiently high spatial resolution for visual identification.

### B. Gaussian Process Classifiers

Most GP classifiers are formulated using a single (latent) output representing a binary classification problem. When faced with multiple classes, it is standard practice to extend the framework using a one-vs-all approach by defining binary models for each class [4], [14], [15], [16]. This effectively leads to  $C$  models of the  $N$  labels at the cost of  $\mathcal{O}(CN^3)$  to invert the covariances.

In problems with large  $N$ , additional algorithms are needed to select a smaller training set [2], or to make solving the covariance matrix tractable through sparsity [17].

In problems with large  $C$ , it has also been recognised that a one-vs-all approach is costly. The redundancy is most clearly illustrated when a standard multi class classifier is applied to a binary problem. In this case the classifier uses two binary models instead of generalising to the normal single-model case. Related work has addressed this problem by modeling  $C - 1$  outputs to imply the missing output [16], [18]. This paper proposes a more compact multi-class generalisation that captures the problem in a single model over the training inputs, rather than repeatedly changing the model for each set of class outputs. This reduces the computational scalability of the approach from  $\mathcal{O}(CN^3)$  to  $\mathcal{O}(N^3)$ .

## III. APPROACH

This section outlines the implementation of the autonomous vegetation classification framework, including image segmentation, the definition of an appearance descriptor vector, and the new extension of the GP classifier to multi-class problems. The following processing steps are currently implemented offline, with the UAV logging imagery and navigation data to a hard drive for post-flight processing.

### A. Feature Space

1) *Colour Balance*: A simple white balance algorithm was devised to improve the robustness of the approach to changes in illumination over the course of a flight. Each RGB image is used to produce an indexed colourmap by minimum variance quantization. The light source is then approximated by fitting a linear relationship between the channels in the quantized colour palette to ensure that large objects of a particular colour will not dominate the result.

2) *Colour*: The use of RGB values is an inappropriate choice for our vision approach due to the poor separation of colour and lighting intensity. Instead, RGB pixel values were transformed to the LUV colourspace, which attempts uniformity over human colour perception.

3) *Texture*: Generic texture filter banks were applied to the luminance channel of the imagery using spectral convolution. Because the trees themselves are non-oriented, rotationally isotropic filters were applied. These consisted of the Schmid (S) rotationally invariant filter bank [19], which contains 13 isotropic, Gabor-like filters, and the Maximum Response Root Filter Set (RFS) filters [20]. The RFS set consists of 38 filters (36 oriented) of which the maximum response of each orientation is recorded. Code to extract these 8 features was provided with the publication [20].

4) *Pre-segmentation*: Because colour and texture responses are noisy from pixel to pixel, the images were pre-segmented into super-pixels containing at least 100 pixels. This was achieved using the fast, unsupervised mean-shift clustering [21], with edge detection provided by [22]. An example of image pre-segmentation is shown in Figure 2.

5) *Linear Dimensionality Reduction*: The GP classifier learns a length scale hyperparameter between every feature attribute and every class. Having 4 classes already, the 3 colour features and 21 texture features lead to an extremely large 96 degrees of freedom that is problematic for numerical



Fig. 2. Each image frame is pre-segmented using an unsupervised mean-clustering implementation from the literature [21]. This procedure splits each tree crown into many super-pixel segments for further processing.

optimisation in terms of convergence (avoiding local minima) and the selection of initial conditions. A multi-class Linear Discriminant Analysis [23] was used to compress the feature attributes into 4 dimensions and was found to improve learning stability and subsequent performance of all the GP classifiers. Compressing the input dimensions does provide some assistance to the classifier by ensuring that all dimensions are relevant, but we argue that firstly all the models compared have had the same degree of assistance, and secondly, the GPs are still faced with a complex relevance determination problem with 16 degrees of freedom. Colour and texture features were separately reduced to two dimensions each to provide a total of four feature attributes rather than the maximum 3 that can be provided by LDA in a 4 class problem, and to ensure that any relationships involving both colour and texture are handled by the nonlinear GP method rather than the LDA.

### B. GP Learning and Classification

1) *Preliminaries:* This section introduces the basic notation and equations involved in GP inference, prior to their extension and modification in Section III-B.3. A GP output  $f$  is inferred over a  $D$  dimensional input space at  $M$  query points  $\{x^*\} = \{x_i^*\}_{i=1}^M \in \mathbb{R}^D$  by conditioning a Gaussian prior on  $N$  training input/output pairs  $\{X, y\} = \{x_i \in \mathbb{R}^D, y_i \in \mathbb{R}\}_{i=1}^N$ . By assuming a *zero mean*, the positive definite *covariance function*  $K(x_a, x_b)$  is used to generate the multi-variate prior covariance over  $f$  and  $f^*$  jointly:

$$y = f(x) + \varepsilon, \quad \varepsilon \sim N(0, \sigma_n^2) \quad (1)$$

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, x^*) \\ K(x^*, X) & K(x^*, x^*) \end{bmatrix}\right) \quad (2)$$

Eq. 2 is analytically conditioned on the observed training outputs  $y$  to obtain an unbiased, probabilistic estimate of  $f^*$ :

$$f^* \sim N(\mu^*, \Sigma^*) \quad (3)$$

$$\Sigma = [K(X, X) + \sigma_n^2 I] \quad (4)$$

$$\mu^* = K(x^*, X)\Sigma^{-1}y \quad (5)$$

$$\Sigma^* = K(x^*, x^*) - K(x^*, X)\Sigma^{-1}K(X, x^*) \quad (6)$$

This Bayesian formulation also provides a closed form marginal likelihood. By parametrising  $\Sigma$  by *hyperparameters*  $\theta$ , GP learning may be conducted by optimising Eq. 7 with

respect to  $\theta$ . This procedure is naturally resilient to overfitting because it is penalised by both data fit and model complexity.

$$\log(p(y|X, \theta)) = -\frac{1}{2}y^T \Sigma^{-1}y - \frac{1}{2} \log |\Sigma| - \frac{N}{2} \log 2\pi \quad (7)$$

2) *GP Classification:* A GP is normally used for regression problems, so a GP classifier takes additional steps to deal with categorical data. In a classification problem with  $C$  classes, the  $N$  examples  $\{X_i, y_i | i = 1 \dots N\}$  are assigned categorical outputs  $y_i \in [1 \dots C]$ . At this point, GP *latent outputs*  $f_n^c$  are defined to indirectly model the probabilities of class  $c$  occurring at  $x_n$ . For a multi class problem,  $f$  contains outputs for each of the  $C$  classes, over each of the  $N$  training points, becoming a vector of length  $CN$ :

$$f = (f_1^1, \dots, f_N^1, f_1^2, \dots, f_N^2, \dots, f_1^C, \dots, f_N^C)^T \quad (8)$$

The class probabilities are obtained by convolving the latent outputs  $f^*$  through a sigmoid that squashes them into valid probabilities that are both exclusive (adding to one), and lie between zero and one. We have used the standard softmax (multinomial logistic) function:

$$p(y_i = c | x, f^1, \dots, f^C) = \frac{\exp(f_i^c)}{\sum_{j=1}^C \exp(f_i^j)} \quad (9)$$

The sigmoidal transformation of  $f^*$  into predictive probabilities implies a non Gaussian relationship between  $y$  and  $f$ , which is not analytically tractable within the GP framework. To condition the prior on  $f$ , a variety of sophisticated approximations may be used, including Monte Carlo sampling[14], the Laplace approximation[24], and Expectation propagation[25]. Of these methods, Expectation Propagation is considered state of the art, approximating the GP posterior with a product of local Gaussian likelihoods for each training case [25].

However, it has been argued that a very accurate approximation of the training likelihoods is not critical for the generation of correct predictions [4]. The Probabilistic Least Squares Classification (PLSC) approach used in this work also approximates the training cases as local Gaussian distributions, but uses fixed positive and negative targets (Eqn. 10). This approach is computationally faster and conceptually more straightforward to implement than the alternatives mentioned above, while in practice the predictive performance is normally unharmed [3], [4], [26]. When using this approximation, one additional step must be taken: because the sigmoid is decoupled from the GP learning criterion, it must also be parametrised and learnt to ensure the outputs are well calibrated probabilities.

$$f_i^c = \begin{cases} +1, & y_i = c \\ -1, & y_i \neq c \end{cases} \quad (10)$$

3) *New Multi-Class Formulation:* In multi-class problems with  $C$  classes, it is a standard practice to convert the class labels into  $C$  binary one-vs-all problems, and model each problem separately with its own covariance function[15], [14], [16], [4]. Training and the initial solving of the classifier

under this formulation scales as  $\mathcal{O}(CN^3)$ . However, we point out that the  $C$  binary problems are in fact very similar because the inputs  $X$  remain unchanged in each case, while the binary targets  $f$  are derived from the same labels  $y$ . In fact, the only reason for using  $C$  different models is to allow each class to have its own set of hyperparameters.

In this section, we propose a new generalisation of the GP classifier that compacts the  $C$  different class models into a single covariance over the  $N$  categorical inputs (instead of the  $NC$  repeated binary inputs), leading to a model that scales at  $\mathcal{O}(N^3)$ . This generalisation is based on a subtle difference in interpretation of the problem.

In the standard formulation, the covariance function  $K_a$  (describing class  $a$ ) is applied over all the inputs of *latent function*  $a$  regardless of their training label. To predict a different latent output we need to generate a new model that uses a different covariance function. In our new formulation,  $K_a$  is applied between *training labels* of class  $a$ , regardless of which latent function we are modeling. In addition, covariances between labels of different classes are specified by a suitable cross-covariance function  $K_{ab}$ . This construction leads to a repeated covariance model for each latent output - only  $f^c$  changes.

To formalise this idea, without loss of generality suppose that the training labels  $(X, y)$  were sorted by class, so that  $y$  is a monotonically increasing vector. This will allow the representation of  $K(X, X)$  as a  $C \times C$  blockwise matrix with block  $(i, j)$  corresponding to interactions between labels of class  $y = i$  and labels of class  $y = j$ :

$$K(X, X) = \begin{bmatrix} K_{11} & \dots & K_{1C} \\ \vdots & \ddots & \vdots \\ K_{C1} & \dots & K_{CC} \end{bmatrix} \quad (11)$$

$$y = [1 \ 1 \ \dots \ 2 \ \dots \ \dots \ C]^T \quad (12)$$

The  $K$  matrix ( $N \times N$ ) describes the covariance over inputs  $X$ , regardless of which latent output we are modeling. Each  $K_{ii}$  is generated using the covariance function for class  $i$ , and uses hyperparameters  $\theta_i$ . Covariance terms between labels of different classes are provided by a cross covariance function  $K_{ij}$ ,  $i \neq j$  that is derived from  $K_{ii}$  and  $K_{jj}$ , and consequently depend on hyperparameters  $\theta_i$  and  $\theta_j$ . It is necessary to include these  $K_{ij}$  terms to handle the negative values in  $f^c$ . Suitable cross covariance functions already exist in the GP literature, because multi-task GPs place a joint covariance function of the same form as Eq. 11 over multiple outputs to learn their dependencies[27]. In this work, we define an exponential covariance function in terms of the diagonal squared length scale hyperparameter matrix  $\lambda$ :

$$K_{ii}(x_a, x_b | \lambda_i) = \exp [-(x_a - x_b)^T \lambda_i^{-1} (x_a - x_b)]^{\frac{1}{2}} \quad (13)$$

$$K_{ij}(x_a, x_b | \lambda_{i,j}) = \exp [-(x_a - x_b)^T (\lambda_i + \lambda_j)^{-1} (x_a - x_b)]^{\frac{1}{2}} \quad (14)$$

A signal amplitude is also applied to  $K$  to specify how tightly the model fits the data. This is introduced into the framework as gain hyperparameters  $\sigma_c^f$  such that the resulting model for latent output  $c$  is given by  $\Sigma^f = \sigma_c^f K(X, X)$ .

While this technically leads to a different  $\Sigma$  for each output, we can take a numerical shortcut by specifying the inverse of  $\Sigma^f$  in terms of  $K(X, X)$  and its inverse:

$$\Sigma^{f^{-1}} = (\sigma_c^f K(X, X))^{-1} = \frac{K(X, X)^{-1}}{\sigma_c^f} \quad (15)$$

The marginal likelihood of  $y$  given  $K$  can be evaluated by multiplying the marginal likelihoods of each  $f^c$  (derived from  $y$ ). In log-form, this becomes a summation over the  $C$  binary cases. In addition, by applying the relationship in Eq. 15, the full marginal likelihood can be expressed in terms of  $K(X, X)^{-1}$ :

$$\log p(y|X, K, \sigma^f) = \sum_{c=1}^C \left( \frac{-1}{2\sigma_c^f} f^{cT} K(X, X)^{-1} f^c - \frac{N}{2} \log \sigma_c^f \right) \dots - \frac{C}{2} \log |K(X, X)| - \frac{NC}{2} \log 2\pi \quad (16)$$

The cost of the Bayesian GP learning is now dominated by the inversion of  $K(X, X)$ , with a scalability of  $\mathcal{O}(N^3)$ . The learning optimisation is conducted using a standard gradient descent routine on the negative log marginal likelihood. We have also found that in practice, introducing a small penalty for when the length scales of each class become very different helps the optimisation avoid sub-optimal local minima.

After inverting  $K(X, X)$ , we can use  $K(X, X)^{-1}$  to predict the  $C$  latent outputs for a query point at trivial additional cost. However, because we are using a multi-task covariance function,  $K(x^*, X)$  depends on the class of the query point  $x^*$ . An effective approach here is to query each latent output with an  $x^*$  belonging to the same class, which we have written as  $x_c^*$ . This leads to the following prediction equations for latent output  $f_c^* \sim N(\mu^*, \Sigma^*)$  corresponding to output function  $c$  at input location  $x^*$ :

$$\mu_c^* = K(x_c^*, X) K(X, X)^{-1} f^c \quad (17)$$

$$\Sigma_c^* = \frac{K(x_c^*, x_c^*) - K(x_c^*, X) K(X, X)^{-1} K(X, x_c^*)}{\sigma_c^f} \quad (18)$$

## IV. RESULTS

### A. Experimental data

1) *Data acquisition*: The capabilities of the terrain classification algorithm were investigated in relation to the identification of two invasive woody weed (tree) species over a farmland site in Northern Queensland, Australia. The UAV was deployed in the field to acquire high resolution colour imagery using a pre-programmed flight path targeting the high vegetation concentration found along a known (seasonally dry) riverbed. The autonomous flight processed in this paper was approximately one hour long and covered 100km of swath-track along a  $3.5 \times 0.6$  kilometre region. During this flight, the payload computer logged navigation sensor data and recorded time-stamped  $1024 \times 768$  images at 3.75 frames per second, a rate designed to give consistent frame overlap for full coverage along swaths.

A region of interest consisting of 1158 images, and covering an area approximately  $1500 \times 400$  metres, was selected from the available dataset.



Fig. 3. A visualisation of some of the manual labels provided to the classifier, with colour overlays indicating segments that have been labeled for each class. A total of 160 such sparse samples (40 of each class) were drawn over 32 manually selected frames to build a training set.

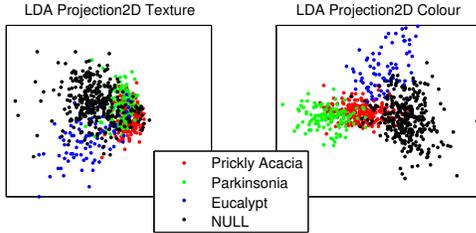


Fig. 4. Multi-class LDA is used to compress the input attributes space to 4 dimensions, allowing the classifier to learn 16 length scale hyperparameters (4 dimensions over for 4 classes).

2) *Identification of Training Examples:* Within the region of interest, 32 frames were selected as being characteristic of the dataset. These frames were selected over the whole dataset, but were not overlapping each other to avoid contamination of the validation data. A human was provided with an interface to identify types of vegetation in the imagery, cross referenced with a ground based-survey of a small portion of the operating area using a handheld GPS receiver. A balanced training set was formed by using 40 examples of each class, leading to 160 training labels overall. Examples were drawn from different segments of the same tree crown in some cases, but this is necessary so that the classifier can learn different shading and patterns as they vary over a typical tree crown. An example of a manually labelled frame is depicted in Fig. 3. A visualisation of the LDA compressed training data is shown in Fig. 4.

### B. Classifier Training and Performance

Given the  $(X, y)$  specified by the manual labeling, the GP classifier model was trained by numerically optimising the log marginal likelihood criterion in Eq. 16. Following learning, 10 fold cross validation was conducted over the training data to evaluate the classifier’s predictive performance. A confusion matrix was built by taking the highest predicted output as the decision (although this does ignore uncertainty), yielding a performance accuracy of  $0.88 \pm 0.06$ . This confusion matrix, and the related one-vs-all precision and recall statistics for each class (true positives / all positives, and true positives / all occurrences respectively) are shown in Table II. The confusion matrix shows good performance for the validation (particularly considering this is a 4 class problem).

We have shown theoretically that our classification approach scales better than the standard GP classifiers. It is also

TABLE I  
CONFUSION MATRIX, PRECISION AND RECALL

		Predicted Class			
		PA	PK	EUC	NULL
Actual Class	PA	36	2	0	2
	PK	6	34	0	0
	EUC	0	0	39	1
	NULL	4	1	3	32

TABLE II  
PRECISION AND RECALL

New multi-class approach: $88\% \pm 6\%$ Accuracy		
	Precision	Recall
PA	0.78	0.90
PK	0.92	0.85
EUC	0.93	0.98
NULL	0.91	0.80

based on the computationally inexpensive probabilistic least squares classifier, which is already significantly faster than the EPGP classifier that is considered the state of the art in terms of predictive performance. However, the performance concessions made to achieve this are not significant. We have also trained a standard PLSC classifier, and an EPGP classifier to investigate the relative performance. The results are summarised in Table III.

TABLE III  
PERFORMANCE OF BENCHMARK CLASSIFIERS

Probabilistic Least Squares: $87\% \pm 9\%$ Accuracy			Expectation Propagation: $89\% \pm 6\%$ Accuracy		
	Precision	Recall		Precision	Recall
PA	0.79	0.85	PA	0.79	0.85
PK	0.86	0.90	PK	0.88	0.90
EUC	0.92	0.93	EUC	0.95	0.95
NULL	0.91	0.80	NULL	0.94	0.85

Comparing the classifier performances, it is clear that while the state-of-the-art EPGP has achieved the highest prediction accuracy, the margins between this approach and both our modified least squares classifier and the standard PLSC classifier, are very small. If we were to introduce additional classes and additional training labels in the future, it would be hard to justify training an EPGP classifier when we are able to achieve essentially equal performance from our fast, computationally inexpensive approach.

### C. Prediction and Mapping

The GP model with optimised hyperparameters was queried over all the segments of all the images in the region of interest. Following this, the predicted class probabilities for the three tree classes were rendered back onto the image segments to create a classification image with red indicating Prickly Acacia (PA), green indicating Parkinsonia (PK), and blue indicating Eucalypt (EUC). Spatial continuity is exploited to de-noise the classifications with a median filter. Figure 5 shows an example of a successful classification for an image frame with all classes present.

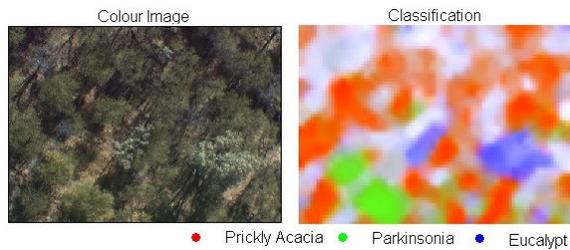


Fig. 5. Probabilistic classification over an entire image frame. The Null background class is rendered white, while PA, PK and EUC are rendered in the red, green, and blue channels respectively such that a stronger colour indicates a higher predictive probability.

Close inspection of the classified image frames reveals why there is some confusion between the null class and the trees. It is likely that these errors stem from the misidentification of shadows (that were defined as part of the null class). Interestingly, the woody weeds and their shadows possess a very similar appearance in this data. They both have relatively dark shading, and the shadow inherits size and texture from the tree. This suggests an additional input attribute such as height would clarify much of this confusion. An example of this occurrence is depicted in Figure 6.

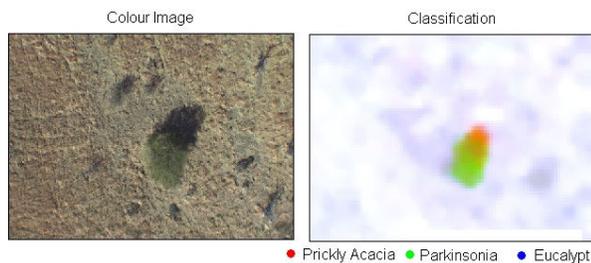


Fig. 6. An example depicting the most common type of incorrect classification, where the shadow immediately adjacent to a tree (in this case, Parkinsonia) has been incorrectly labeled as prickly acacia.

The classified frames were integrated with the navigation solution of the UAV to render a map of the region of interest. A detailed treatment of the navigation process is described in [5]. Figure 7 shows the white balanced colour input imagery over the region of interest, together with the corresponding classification visualisation. As we already knew (from ground surveying), the invasive Prickly acacia (red) has extensively infested this region of farmland. The map also reveals scattered individuals of surviving native Eucalypt (blue), competing with the less successful Parkinsonia weed (green), but primarily in densely populated regions close to the center of the watercourse.

## V. CONCLUSIONS AND FUTURE WORK

A framework has been presented to classify high resolution colour aerial imagery acquired by our low flying UAV. Generic colour and texture descriptors were derived from the imagery, and these features were averaged over super-pixels obtained by fast unsupervised clustering, and dimensionality reduced using a linear method.

A machine learning classifier was trained to transform the prepared input features into predictive probabilities of each class. For this purpose, we present a new multi-class generalisation of the Gaussian Process classifier. The new generalisation removes the redundancy of the standard one-vs-all approach by packing the underlying models of each class into a single covariance function over the training labels, rather than using multiple binary models. This leads to a more compact covariance representation that allows faster learning, without sacrificing the model flexibility that GP learning provides because length scale hyperparameters are still included to model the relevance of each input dimension to each class type. The new formulation has been beneficial in learning the classifier model (prior to querying the classifier on the image data), and will be well suited to the addition of further classes in new datasets.

We have tested the approach by deploying the UAV over a survey site in Northern Queensland, Australia, to collect imagery along a woody weed infested river system. Having manually labeled just a fraction of the total amount of imagery (40 segment labels per class), the new GP formulation was trained, and its performance was measured quantitatively (by cross validation) and qualitatively by producing a class probability map. Cross validation on the training data has indicated accuracies of up to 88%, a good result considering the difficulty of the 4 class problem. The classifier has been able to learn a complex model that would be difficult to tune manually (without prior knowledge), owing to the elegant marginal likelihood based learning that a GP framework provides.

We note that the primary sources of confusion in our results are between the two similar invasive trees, and also between the trees and their shadows on light coloured ground. This second problem may be solved in future work by replacing the unsupervised segmentation with a shadow or height based segmentation. Stemming from such an approach, classification reliability may be further improved if we are able to incorporate features such as shape and tree crown size into the description vector.

## REFERENCES

- [1] R. Lawes and J. Wallace, "Monitoring an invasive perennial at the landscape scale with remote sensing," *Ecological Management and Restoration*, vol. 9(1), pp. 53–58, 2008.
- [2] N. Lawrence, M. Seeger, and R. Herbrich, "The informative vector machine: a practical probabilistic alternative to the support vector machine," Dept. Computer Science, University of Sheffield, Tech. Rep., 2005.
- [3] H. Nickisch and C. Rasmussen, "Approximations for binary Gaussian process classification," *Journal Machine Learning Research*, vol. 9, pp. 2035–2078, 2008.
- [4] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [5] M. Bryson, A. Reid, F. Ramos, and S. Sukkarieh, "Airborne vision-based mapping and classification of large farmland environments," *Journal of Field Robotics*, vol. 27 (5), pp. 632–655, 2010.
- [6] B. Sofman, A. Bagnell, A. Stentz, and N. Vandapel, "Terrain classification from aerial data to support ground vehicle navigation," Robotics Institute Carnegie Mellon University, Tech. Rep., 2006.
- [7] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert, "Natural terrain classification using 3-d LADAR data," in *Proc. IEEE Int. Conf. Robotics Automation*, 2004.

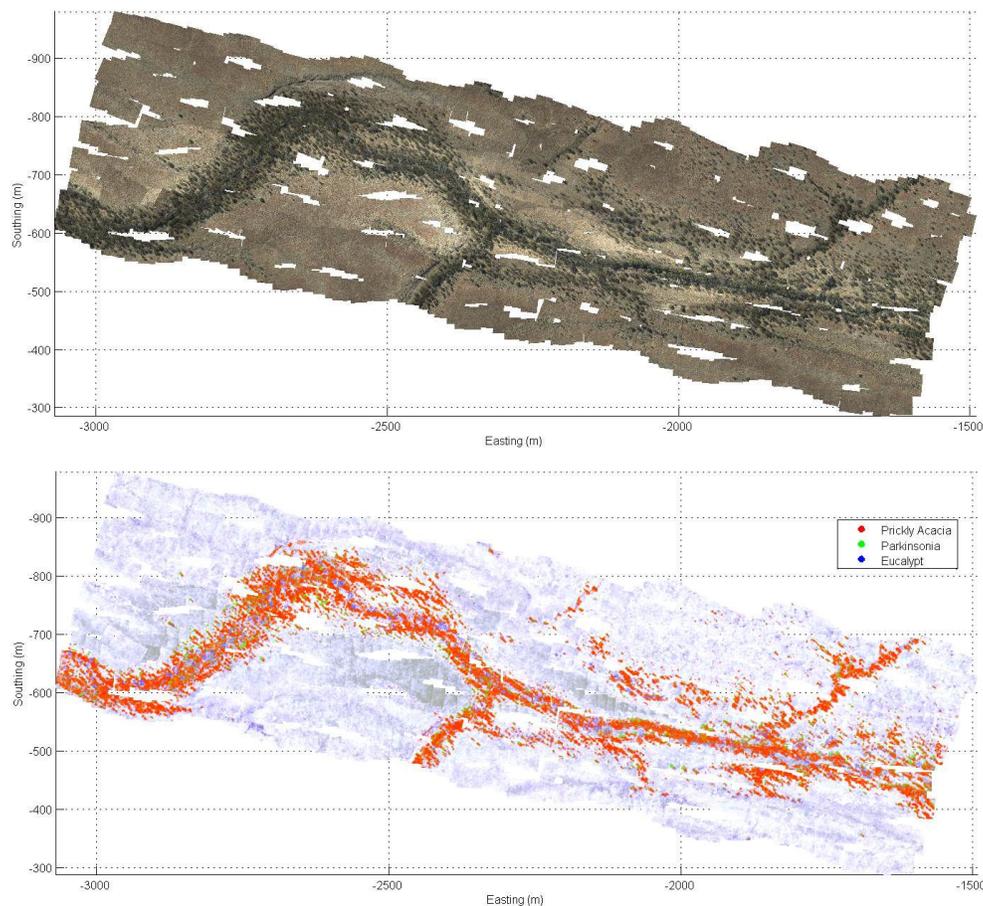


Fig. 7. Top: Map of corrected imagery over the region of interest (1158 individual frames). Bottom: Corresponding probabilistic classification. The Null background class is rendered white, while PA, PK and EUC are rendered in the red, green and blue channels respectively such that a stronger colour indicates a higher predictive probability. Coordinates are given in metres in a local North-East frame.

- [8] Q. Yu, P. Gong, N. Clinton, G. Biging, M. Kelly, and D. Schirokauer, "Object-based detailed vegetation classification with airborne high spatial resolution remote sensing imagery," *Photogrammetric Eng. Remote Sensing*, vol. 72(7), pp. 799–811, 2006.
- [9] C. Medlin, D. Shaw, P. Gerard, and F. LaMastus, "Using remote sensing to detect weed infestations in glycine max," *Weed Science*, vol. 48(3), p. 393398, 2000.
- [10] X. Ye, K. Sakai, S.-I. Asada, and A. Sasao, "Use of airborne multispectral imagery to discriminate and map weed infestations in a citrus orchard," *Weed Biology and Management*, vol. 7(1), pp. 23–30, 2007.
- [11] L. Tang, L. Tian, and B. Steward, "Classification of broadleaf and grass weeds using Gabor wavelets and an artificial neural network," *Trans. ASAE*, vol. 46(4), pp. 1247–1254, 2003.
- [12] T. Cox, C. Nagy, M. Skoog, and I. Somers, "Civil UAV capability assesment," NASA, Tech. Rep., 2004.
- [13] A. Rango, A. Laliberte, C. Steele, K. Herrick, B. Bestelmeyer, T. Schmutge, A. Roanhorse, and V. Jenkins, "Using unmanned aerial vehicles for rangelands: Current applications and future potentials," *Journal Environmental Practice*, vol. 8, pp. 159–168, 2006.
- [14] M. Girolami and S. Rogers, "Variational Bayesian multinomial probit regression with Gaussian process priors," *Neural Computation*, vol. 18(8), pp. 1790–1817, 2006.
- [15] M. Andriluka, L. Weizsäcker, and T. Hofmann, "Multi-class classification with dependent gaussian processes," in *12th Int. Conf. Appl. Stochastic Models Data Anal.*, 2007.
- [16] H.-C. Kim and Z. Ghahramani, "Bayesian Gaussian process classification with the EM-EP algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28(12), pp. 1948–1959, 2006.
- [17] A. Melkumyan and F. Ramos, "A sparse covariance function for exact gaussian process inference in large datasets," in *21st Int. Joint Conf. Artificial Intell.*, 2009.
- [18] R. Neal, "Regression and classification using Gaussian process priors," *Bayesian Statistics*, vol. 6, pp. 475–501, 1997.
- [19] C. Schmid, "Constructing models for content-based image retrieval," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 2, 2001, pp. 39–45.
- [20] J. Geusebroek, A. Smeulders, and J. Weijer, "Fast anisotropic Gauss filtering," *IEEE Trans. Image Process.*, vol. 12(8), pp. 938–943, 2003.
- [21] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, 2002.
- [22] P. Meer and B. Georgescu, "Edge detection with embedded confidence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, pp. 1351–1365, 2001.
- [23] Martinez and Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23(2), p. 228233, 2004.
- [24] C. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20(12), pp. 1342–1351, 1998.
- [25] T. Minka, "Expectation propagation for approximate Bayesian inference," in *Uncertainty Artificial Intell.*, 2001, pp. 362–369.
- [26] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with Gaussian processes for object categorization," in *Proc. Int. Conf. Comput. Vision*, 2007.
- [27] E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams, "Multi-task Gaussian process prediction," in *Advances Neural Inform. Process. Syst. 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. MIT Press, 2008.