2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Classifying Natural Aerial Scenery for Autonomous Aircraft Emergency Landing

Luis Mejias

Abstract— **In this paper, we present an approach for image-based surface classification using multi-class Support Vector Machine (SVM). Classifying surfaces in aerial images is an important step towards an increased aircraft autonomy in emergency landing situations. We design a *one-vs-all* SVM classifier and conduct experiments on five data sets. Results demonstrate consistent overall performance figures over 88% and approximately 8% more accurate to those published on multi-class SVM on the KTH TIPS data set. We also show per-class performance values by using normalised confusion matrices. Our approach is designed to be executed online using a minimum set of feature attributes representing a feasible and ready-to-deploy system for onboard execution.**

I. Introduction

In the forthcoming years aircraft autonomy will play a decisive role in the integration of unmanned aircraft (UA) in the civilian airspace. Unmanned aircraft are deemed to demonstrate equivalent levels of performance and safety to that of manned counterparts. Particularly, in scenarios where a human pilot has been the major player in resolving emergency situations (e.g emergency landing). This is a critical capability required if unmanned aircraft are intended to perform routinely tasks in populated areas. Currently, there are a number of contributions aiming at resolving optimally challenges in control, localisation, guidance & navigation, etc; in a variety of indoor and outdoor environments. However, little attention has been placed in providing the necessary levels of autonomy to resolve emergency situations, such as emergency landing. Our problem has the same magnitude to the one faced by many roboticist when designing robots for safe human-robot interaction.

The most commonly employed method to diminish the severity of an UA emergency landing is the use of parachutes or parafoils [1]. Whilst this concept is attractive in that it still enables limited vehicle controllability even when both the engine and control surfaces have failed, it is highly susceptible to wind gusts and other atmospheric effects which may adversely affect the final impact point. Furthermore, the potential impact on the ground and in public perception is still significant if we consider an aircraft of several kilos descending and impacting on private property.

In this context, is where our project stands in such a way that it provides algorithms that resemble human behaviour allowing an aircraft with some degree of controllability to autonomously localise a landing site on the ground and then guide the aircraft towards it [2]. In particular, this paper deals with the problem of localisation of landing sites as seen by an onboard camera [3]. Vision as a sensor not only offer a rich source of information for navigational purposes, but it also offer the best chances for regulator approval, at the time it addresses limitations such as available onboard power, size and weight. Furthermore, the maturity of computer vision is at the stage that can be used in real scenarios. This work is integrated in a novel end-to-end multi-layered architecture that detects (2D segmentation [4], Figure 1), reconstruct (3D [5]) and classify (this work) aircraft landing sites using images from an onboard monocular camera.

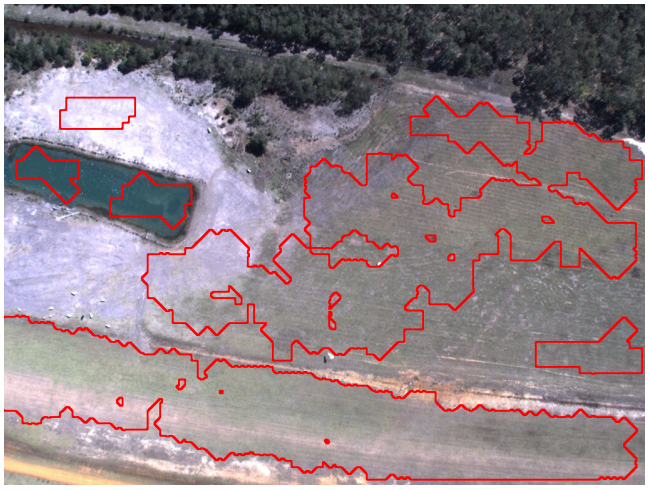

Fig. 1: Typical output of our candidate landing site selection algorithm [4]. Areas within the red contour have been segmented as possible landing sites.

In this sense, this paper presents an algorithm part of a multi-layered framework (Figure 2) to visual landing detection and classification. Our algorithm uses multi-class Support Vector Machine (SVM) to classify areas in images that correspond to different types of ground surface (e.g bitumen, water, grass, trees and buildings). We introduce the algorithm and its test outcomes on realistic visual data collected from an aircraft.

The main differentiating point of this work lies in the accuracy and ability to work online in a cascade type implementation as depicted in Figure 2. While training still occurs offline, testing is performed analysing sequentially images from several data sets. Furthermore, due to the nature of the problem in hand and the low number of classes this work does not address scenarios in which new classes (unmodelled) can be actively learnt or classified with some degree uncertainty, rather we focus on designing an algorithm

Luis Mejias is with the Australian Research Centre for Aerospace Automation at Queensland University of Technology Brisbane, Australia luis.mejias@qut.edu.au

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

with a high degree of confidence that can be executed online in near real time. Consequently, our work inherits many features of standard machine learning techniques, however making the following contributions: i) A thorough design of a multi-class SVM achieving significant performance increase when compared with previous published work using standard benchmark data sets [6]. ii) Online cascaded implementation by leveraging in a novel multilayered approach for near real-time performance. iii) Consistent overall performances over 88% using four realistic data sets at various altitudes with a minimum set of feature attributes.

This paper is structured as follows. Section II presents an brief overview of recent work on machine learning classification. Section III describes the main approach to multi-class SVM, feature selection and classification. Section IV outlines the experimental design and data analysis. Finally, section V describes some of the lessons learnt and future work planned.

II. Background

The work presented here follows that of previous research that is looking at the suitability of colour vision techniques for the automatic selection of landing sites for aircraft performing an emergency landing. Previous work has looked at size, shape and slope determination, which produces a result similar to that shown in Figure 1. Vital however to the final site selection process is surface type classification, hence the work presented here which addresses the problem of surface classifier design.

Texture classification is a fundamental problem in computer vision and its of paramount importance in a wide range of applications that includes object recognition and segmentation, medical imaging, remote sensing and many more. In our problem, textures are classified to categorise the type of surface underneath the aircraft. This is important to define attributes for a landing site that are then used to prioritise the areas that are suitable for an aircraft emergency landing.

A large body of work already exist in supervised learning methods. Traditionally, terrain or surface classification has been addressed by the remote sensing community [7]. Techniques such as Artificial Neural Networks (ANN), Gaussian Mixture Models (GMM), AdaBoost, Random Forrest, Support Vector Machine (SVM), etc have seen wide use in terrain classification [8], [9], [10], [11]. Each have their advantages and disadvantages, however their best results are obtained through the careful selection of input features and appropriate training practices. Recently, several researchers have contributed to the terrain classification problem for ground robot navigation [8], [12] (see references within). Aerial robots have also seen activity in the terrain classification problem. To date, Unmanned Aerial Systems (UASs) have provided benefits in the data collection for image classification, followed in most cases by data post-processing [13][14]. However, the use of classified imagery for realtime navigation and particularly to resolve an emergency situation have not been widely addressed. Recent related work on this domain includes [15], [16], [17].

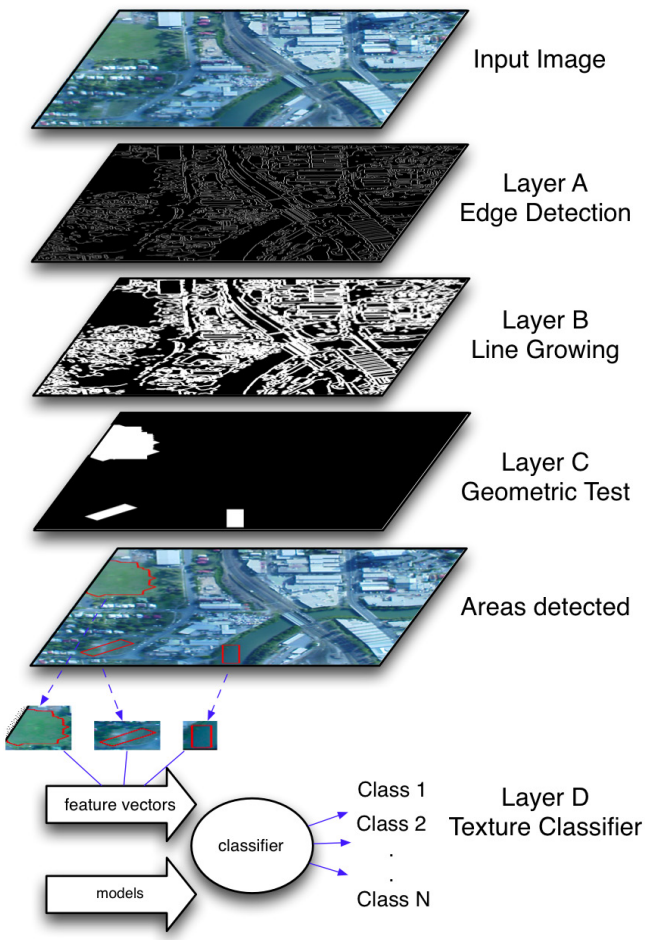

Fig. 2: Algorithm outline showing the four main processing stages. Details of layers A-C can be found in [4].

In a natural way, our work builds upon previous research on machine learning, and particularly on classification using multi-class support vector machine. Since its proposal SVMs have gained an enormous popularity in statistics, learning theory, and engineering (see for instance [18][19], and the many references therein). SVMs have proved particularly promising because of its lower sensitivity to input vector dimensionality compared to traditional classification approaches. The main reason of such an attractive property, is due to the margin maximisation principle they are based on, hence it becomes unnecessary to explicitly estimate the statistical distributions of classes in the hyperdimensional feature space to carry out the classification task. Another important property is their good generalisation capability supported by their sparse representation of the decision function. These advantages have taken SVMs to a consolidated position in pattern recognition [20]. In this paper, instead of following common benchmarking approaches in which several techniques are tested and their merits assessed for a particular problem, we have focused on designing and optimising the best multi-class SVM approach, and comparing

its implementation with published results on public data sets, as well as on real aerial imagery.

III. Methods

In this section we describe the general approach to classification using multi-class SVMs. Our approach aims to address the classification of five types of surfaces namely $Bitumen$, $Water$, $Grass$, $Trees$ and $Buildings$. The approach involves two stages such as feature extraction and classification.

A. Review of Multi class SVM

Let us consider a training data $D = \{(x_1, y_1), ..., (x_l, y_l)\}$ with $x_i \in \Re^n, i = 1, ..., l$ exist from the n-dimensional feature space X. To each component in D we associate a class $y_i \in \{-1, +1\}$. A linear SVM consist of finding the separation between two classes in X by means of an optimal hyperplane that maximises the separating margin. In a nonlinear case, which is the most commonly used, data is first mapped using a kernel function to a higher dimensional feature space $\Phi(X) \in \Re^{n'}(n' > n)$. The membership to a given class is based on the function $sign[f(x)]$, where $f(x)$ represents the discriminant function associated with the hyperplane in the higher dimensional feature space and is defined as $f(x) = w \cdot \Phi(X) + b$ where w is a weight vector and b bias. The optimal hyperplane is found minimising

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\xi_i \tag{1}$$

subject to the following constraints

$$y_i(w \cdot \Phi(X) + b) \geq 1 - \xi_i \ , \ i = 1, 2, ..., l \tag{2}$$

$$\xi_i \geq 0 \ , \ i = 1, 2, ..., l \tag{3}$$

where ξ_i are variables introduced to account for non-separable data. The constant C is the regularisation parameter that allows to control the shape of the discriminant function and, consequently, the decision boundary when data are non-separable. The above optimisation problem can be reformulated through a Lagrange functional for which the Lagrange multipliers (e.g $\alpha = [\alpha_1, \alpha_2, ..., \alpha_l]$) can be found leading to

$$f(x) = \sum_{i=1}^{l}\alpha_i y_i K(x_i, x) + b \tag{4}$$

where $K(.)$ is the kernel function typically represented by the Gaussian function $K(x_i, x) = e^{-\gamma||x_i - x||^2}$ where γ is inversely proportional to the width of the Gaussian kernel. Essentially, a SVM is a two class classifier, to construct a multi-class classifier one should build several classifiers that approach the problem in one the form described next.

B. One-vs-One and One-vs-All approaches

A multi-class SVM can be constructed using one of the following approaches, one-vs-all (OVA) [18], one-vs-one (OVO) [21], ECOC SVM [22] and DAGSVM [23] which are based on binary classifiers. The choice of one approach or the other is highly context dependent. A large body of recent studies have attempted to compare and claim the superiority of one method against the others [24], [25], however no one has conclusively stated the dominance of a particular approach. In fact, it seems that factors such as practical implementation, training speed, accuracy, number of samples/features, etc have all an effect in the adoption of a particular approach. In this work, rather than comparing several classification techniques or methodologies we focus on implementing and optimising an OVA SVM approach, and comparing this implementation with published results on public data set [6]. Classification accuracy between OVO and OVA approaches is not significantly different [24], therefore the choice of technique adopted is related to the uniqueness of the data set used. In the implementation of an OVA approach, N different binary classifiers (one for each class) are built, each one trained to distinguish samples in a single class from samples in all remaining classes. To classify a new sample, the N classifiers are run and the classifier which outputs the largest (most positive) value is chosen. The implementation used here is based on publicly available routines, specifically it uses a set of Matlab® scripts around the C-implementation of libsvm [26], [27].

C. Feature Extraction

The use of appropriate features to characterise an output class is fundamental for any classification problem [28]. Input features are used during the design process in both, training and testing phases hence the importance of this step. Whilst a large number of features that are easy to compute, robust to distortion and illumination changes, rotationally invariant, etc; might be desirable to characterise a given class, in practice large number of features will most likely lead to high computational times in both training and testing. The choice of output classes is also an important component of classifier design, as the classes must encompass the different surfaces that may be encountered by the classifier. If all plausible classes are not represented in the classifier output, a conflict of class identities may occur and result in an overall decrease in the classifier accuracy.

To form a benchmark, a subset of the 35 features were selected which had been previously reported as successful in the development of classifiers [29]. Input feature vectors are also normalised to ensure precedence is not placed on one feature over another. Principal Component Analysis (PCA) is then used to re-balance the weight amongst similar features. This not only helps to reduce noise but results in decreased training time, number of training samples required and overall computation time. Our feature vector contains the following features HSV 1-3: mean; 4-6: variance, Normalised RGB 7-9: mean; 10-12: variance, RGB 13-15: mean; 16-18: variance, Seven Gabor images with seven frequencies each

averaged over 4 orientations; 19-25: mean; 26-32: variance, and 33-35: the blur-insensitive Local Phase Quantization (LPQ) method [30] on H, S and V respectively (Table I). The choice of LPQ was motivated by its insensitivity to image blur which can be significant in aircraft at lower altitudes.

TABLE I: Features used for classification experiments

No.	Feature	Reference
	HSV Colour Space	([31],[32],[33])
1-3	H,S,V Mean	
4-6	H,S,V Variance	
	Normalised RGB Colour Space	...
7-9	R,G,B Mean	
10-12	R,G,B Variance	
	RGB Colour Space	...
13-15	color ratio Mean RG,RB,GB	
16-18	color ratio Variance RG,RB,GB	
	Grey level	[34], [35]
19-25	gabor image mean 7 Freq./scales	
26-32	gabor image variance 7 Freq./scales	
	HSV Colour Space	([30], [36])
33-35	LPQ Variance on H,S,V	

D. Classification

Supervised machine learning tasks often boils down to the problem of assigning labels to instances where the labels are drawn from a finite set of elements. This is done by finding the optimal hyperplane that can divide a set of data points into their true classes. In this optimisation process several kernels such as Radial Basis Function (RBF), Linear, Polynomial or Sigmoid can be used. During experimentation, these kernels produced accuracies comparable to each other, with RBF performing the best (see Section IV-B). This kernel can handle reasonably well nonlinear relationships between class labels and attributes. The general form of a Radial Basis Function (RBF) kernel is $K(x_i, x) = e^{-\gamma||x_i - x||^2},\ \gamma > 0$. However, two important parameters (C, γ) must be optimally found as they are unknown beforehand. This step involves a parameter search until a certain level of accuracy is achieved. In addition, in order to prevent overfitting we perform cross-validation using *5-fold*. We found empirically that growing sequences exponentially for C and γ in range of $C = 2^{K_1}$, with $K_1 = [-7, 7]$ and $\gamma = 2^{K_2}$, with $K_2 = [-10, 4]$ offered best tradeoff between accuracy and training time.

IV. Experimentation and Analysis

A. Data sets and experiment setup

Five data sets containing colour images at different resolutions and frame rates are used in these experiments. Not all images are processed during the experiments, however representative subsamples of the data sets containing different terrain in urban and rural areas are analysed. The first data set (A) contain images captured at 7.5Hz with a resolution of 720x576 pixels at 1000ft. The second data set (B) acquired at 7.5Hz with a resolution of 1024x768 pixels at 4000ft. The third data set (C) captured at 20Hz in a 90min duration flight with a resolution of 1024x768 pixels. The fourth data set (D) captured at 15Hz with a resolution of 720x576 pixels at 1000ft. Finally, the fifth data set (E) is the KTH TIPS [6] data set used during the development of the classifier.

Data set E is selected for classifier development and testing since it provides published benchmarks to compare with. We divide the data set in two subset each containing 405 images of 200×200 pixels for training and testing respectively. Testing images are randomly chosen for 10 different materials (classes), and results with this data set are 98.7% overall accuracy representing $\approx$ 8% performance increase. When compared with published results on SVM classifiers tested on this data set our approach achieved significant performance increase [6], providing a good level of confidence in our design process.

A common challenge when working with real data in unvisited environments is the lack of ground truth. In practice, is unrealisable to label each sample on a set of 78000 images, which is the case of data set C [37]. A combination of several data sets with and without ground truth is often a practical solution to this limitation. During online testing a sixth class ($None$) is defined to deal with the nature of the segmentation layer (see Figures 6 and 7).

In this paper, data sets A and B are made up of image samples of 30×30 pixels with testing classes manually labeled, whereas data set C and D are image samples of 50×50 pixels with testing classes unlabelled. A total of 268 images from data set C and 43 images from data set D were used to test the classifier. Due that these data has no ground truth, performance can only be evaluated if true samples are provided to compared with. We randomly selected 27 images (C) and 5 images (D) to be manually labelled producing 8100 and 900 testing samples respectively. Each full image in data set C and D is divided by the image sample size producing the number of training and testing samples in Table II. Plots shown in this section use colours generated at runtime to differentiate classes rather than identify them.

TABLE II: Data sets used in experiments

Data set	Altitude (ft)	Training	Testing	Classes	Accuracy (%)
A	1000	566	133	5	90.18
B	4000	451	71	5	88.12
C	1000	300	8100	4	92.32
D	1000	204	900	5	96.61
E	na	405	405	10	98.73

B. Analysis

Unbalanced number of samples and classes call for methods that allow seamless comparison between results from different data sets. Here, we normalise each confusion matrix (error matrix) using iterative proportional fitting procedure. In this way, differences in sample sizes used to generate the matrices are eliminated, and individual cell values within the matrix are directly comparable. Furthermore, each cell value can rapidly be converted into a percentage multiplying by 100 (see Tables III & IV). Prior to conducting the experiments we evaluated 3 kernels with data set (E) obtaining the following accuracies Linear: 94.78%, Polynomial: 75.69%, Sigmoid: 89.78% and RBF: 98.73%. The RBF kernel was

then chosen to conduct the experiments with the remaining data sets. Table III shows the classification results for data sets A and B. Directly comparing the classes $Bitumen$ and $Trees$ we observe little difference between the two data sets, however taking into account the rest of the classes the overall accuracy differs by 2% which might be attributed to the differences in resolution and classes present in these data sets (no class $Water$ is present in data set B). Figures 3 and 4 show these classification results. In these two figures, unfilled and filled markers denote data instances from the training and test set, respectively. In addition, filled colours denote class labels assigned by the classifier whereas edge colours denote the true (ground-truth) label. These two plots were created for visualisation purposes by reducing the dimension from 35D to 2D using the Matlab® functions *pdist, cmdscale and scatter*.

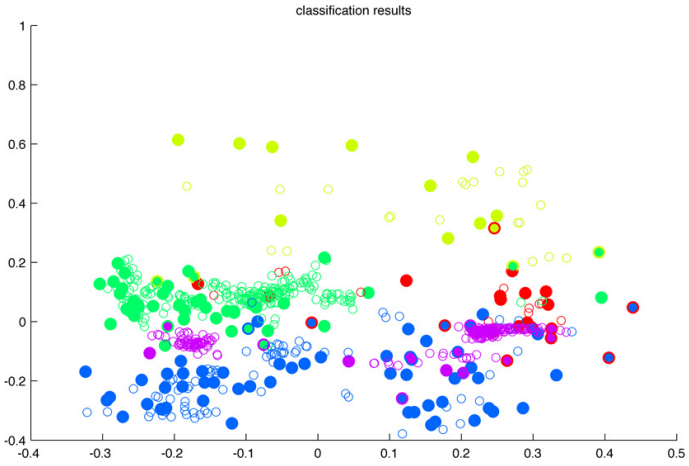

Fig. 3: Classification results for data set A. Plot shows the distribution of input features in the feature space. In this plot, unfilled and filled markers are samples from the training and test set, respectively. Filled colour represent labels assigned by the classifier and edge colour true label.

Classification results for data sets C and D are shown in Table IV. These two data sets differ from previous in that they don't have ground-truth for validation. In order to assess the performance of the classifier on these data sets we manually labelled %10 (randomly selected) of the testing images in each data set, producing 27 error matrices for data set C and 5 for data set D which were then averaged and presented in Tables III & IV. The motivation behind reporting performance on randomised subsample of the total population size is due to: i) the excellence performance achieved during the design phase which provides a degree of high confidence in our implementation, ii) manually labelling 80400 (C) & 7740 (D) which correspond to the total testing population size is would be extremely time consuming.

Examining Table IV, once more we observe excellent performance for class $Grass$ and $Tree$. Figure 5 shows the distribution of input features for these data sets. Examining the features in these two plots one can immediately see how distinctive (or separated) classes are, and therefore have an indication about how arduous the finding of the optimal

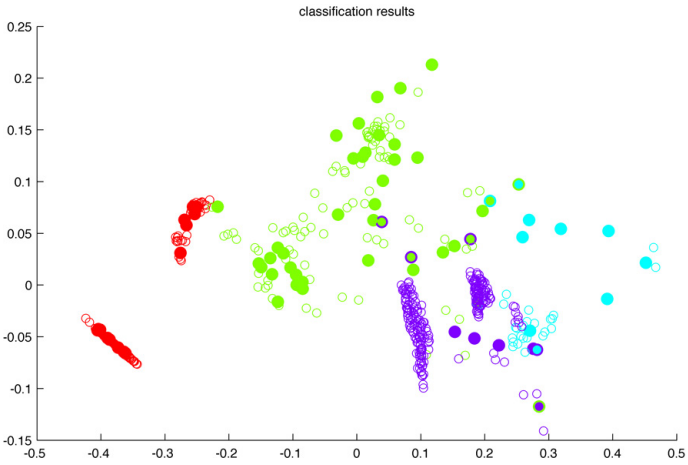

Fig. 4: Classification results of input features distributed in the feature space for data set B. In this plot, unfilled and filled markers denote samples from the training and test set, respectively. Filled colour represent labels assigned by the classifier and edge colour true label.

hyperplane will be. In this scenario, we can observe high overlap in input features for data set C, this situation makes it difficult to find a hyperplane that separates these classes, hence the performances of 88.9%, 85.8% and 93%.

Finally, two example outputs are seen in Figures 6 and 7. Input, segmented and classified image for data sets C and D are shown in these figures. Currently, an area is declared to be of a given class by using winner-take-all approach amongst cells. Final processing rate requirements might require different approaches. However, whilst straightforward winner-take-all provides excellent performance.

TABLE III: Normalised Confusion Matrix: Data set A & B

A	**Bi**	**Bu**	**G**	**T**	**W**
Bi	1.0	0.0066	0	0.0010	0.0021
Bu	0	0.9934	0.010	0	0
G	0	0	0.9157	0	0.0801
T	0	0	0.0139	0.8462	0.1354
W	0	0	0.0604	0.1528	0.7824

B	**Bi**	**Bu**	**G**	**T**	**W**
Bi	1.0	0	0	0	0
Bu	0	0.7955	0.0487	0.1558	0
G	0	0.1147	0.8853	0	0
T	0	0.0898	0.0660	0.8442	0
W	0	0	0	0	0

Bi=Bitumen, Bu=Building, G=Grass, T=Trees, W=Water

V. Conclusions

Designing safe aerial robots is ultimately the aim of our research. By combining several processing stages in an original way we aim to contribute with a novel end-to-end system that can provide possible landing areas to an autonomous aircraft. In this paper, we present an important approach to surface classification using aerial images with consistent overall performance figures over 88%. We also show per-class performances values by using normalised confusion matrices. Our multi-class SVM approach is designed to be

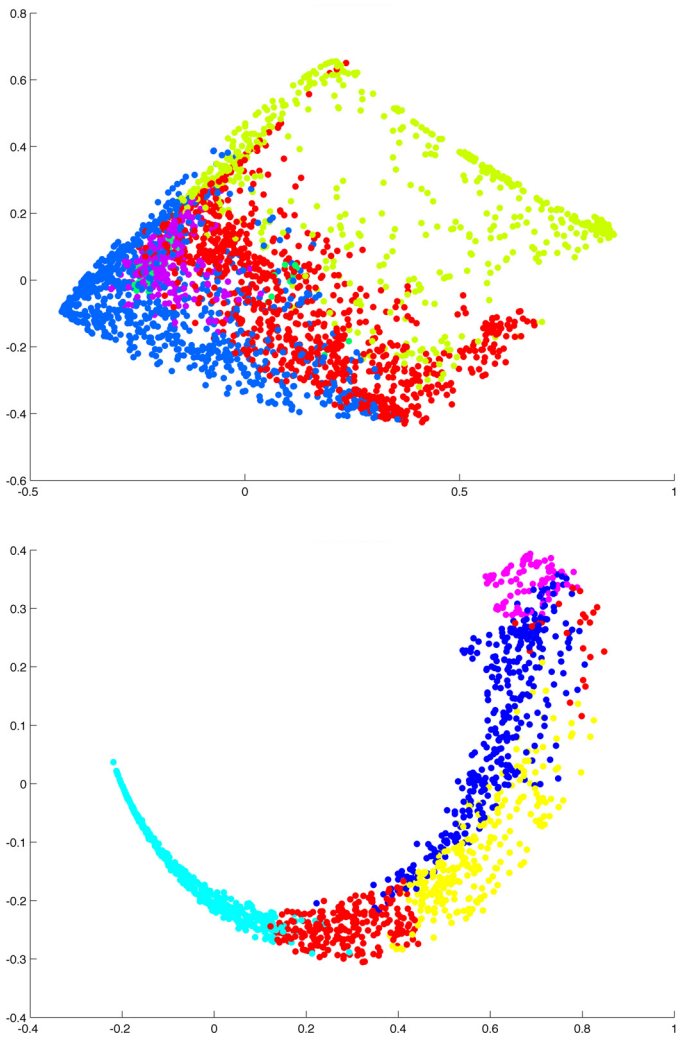

Fig. 5: Input feature vector for samples in data set C (top) and D (bottom). Data set C presents notably more overlap in the input features than data set D.

executed online using a minimum set of feature attributes. By thoroughly designing our approach and achieving a performance increase of $\approx 8\%$ over published benchmarks provide us with confidence levels that are confirmed by randomly subsampling our unlabelled testing data sets. The work presented here is currently under further integration, optimisation and testing within our multi-layered framework.

TABLE IV: Normalised Confusion Matrix: Data set C & D

C	Bi	Bu	G	T	W
Bi	0.8898	0.0.805	0.0297	0	0
Bu	0.11	0.8588	0.0312	0	0
G	0	0.0607	0.9392	0	0
T	0	0	0	1.0	0
W	0	0	0	0	0

D	Bi	Bu	G	T	W
Bi	0.9617	0.03969	0	0	0
Bu	0	0.9694	0.0401	0	0
G	0.01981	0	0.9667	0	0
T	0	0	0	1.0	0.0246
W	0.0272	0	0	0	0.9838

Bi=Bitumen, Bu=Building, G=Grass, T=Trees, W=Water
results shown are average values over 23 and 5 testing images for data set C and D, respectively

REFERENCES

[1] C. Redelinghuys, "A flight simulation algorithm for a parafoil suspending an air vehicle," *Journal of Guidance Control and Dynamics*, vol. 30, no. 3, pp. 791–803, 2007.

[2] L. Mejias, D. L. Fitzgerald, P. C. Eng, and L. Xi, "Forced landing technologies for unmanned aerial vehicles : towards safer operations," in *Aerial Vehicles*, L. T. Mung, Ed. Kirchengasse, Austria: In-Tech, January 2009, pp. 413–440. [Online]. Available: http://eprints.qut.edu.au/42556/

[3] L. Mejias and P. Eng, "Controlled emergency landing of an unpowered unmanned aerial system," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 421–435, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10846-012-9767-5

[4] L. Mejias and D. Fitzgerald, "A multi-layered approach for site detection in uas emergency landing scenarios using geometry-based image segmentation," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, 2013, pp. 366–372.

[5] M. Warren, L. Mejias, X. Yang, B. Arain, F. Gonzalez, and B. Upcroft, "Enabling aircraft emergency landings using active visual site detection," in *9th Field and Service Robotics (FSR). Brisbane*, L. Mejias, P. Corke, and J. Roberts, Eds., 9-11 December 2013.

[6] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds. Springer Berlin Heidelberg, 2004, vol. 3024, pp. 253–266.

[7] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[8] S. T. Namin and L. Peterson, "Classificatoin of materials in natural scenes using multi-spectral images," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, 2012, pp. 1393–1398.

[9] T. Kim, G. Sung, and J. Lyou, "Robust terrain classification by introducing environmental sensors," in *IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, July 2010, pp. 1–6.

[10] J. Chetan, M. Krishna, and C. Jawahar, "Fast and spatially-smooth terrain classification using monocular camera," in *20th International Conference on Pattern Recognition*, 2010, pp. 4060–4063.

[11] X. Miao and J. Heaton, "A comparison of random forest and adaboost tree in ecosystem classification in east mojave desert," in *Geoinformatics, 2010 18th International Conference on*, June, pp. 1–6.

[12] R. Paul, R. Triebel, D. Rus, and P. Newman, "Semantic categorization of outdoor scenes with uncertainty estimates using multi-class gaussian process classification," in *IEEE/RSJ International Conference on Intelligent Robotics and Systems*, October 2012, pp. 2404–2410.

[13] A. S. Laliberte, M. A. Goforth, C. M. Steele, and A. Rango, "Multispectral remote sensing from unmanned aircraft: Image processing workflows and applications for rangeland environments," *Remote Sensing*, vol. 3, pp. 2529–2551, 2011.

[14] F. Hassan, M. MatJafri, and H. Lim, "Contextual classification of cropcam uav high resolution images using frequency-based approach for land use/land cover mapping case study: Penang island," in *Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on*, Sept., pp. 663–668.

[15] Y.-F. Shen, Z. Rahman, D. Krusienski, and J. Li, "A vision-based automatic safe landing-site detection system," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 49, no. 1, pp. 294–311, 2013.

[16] S. Scherer, L. Chamberlain, and S. Singh, "Online assessment of landing sites," in *AIAA Infotech Aerospace*, Atlanta, Apr. 2010.

[17] S. Scherer, L. J. Chamberlain, and S. Singh, "Autonomous landing at unprepared sites by a full-scale helicopter," *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1545–1562, December 2012.

[18] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[19] B. Scholkopf, C. Burges, and A. S. (Eds), *Advances in Kernel Methods - Support Vector Learning*. MIT Academic Press, 1998.

[20] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2,

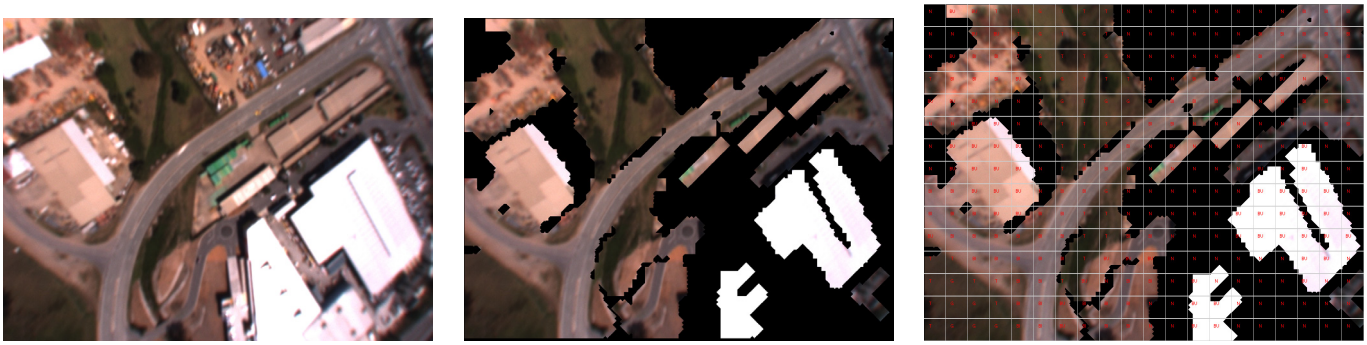

Fig. 6: Classification result on a sample image taken from data set C. Left: input image. Center: segmented image. Right: classification result over an overlaid grid. Six classes shown are BU: building, Bi=bitumen, T=trees, G=grass, W=water and N=none. Each cell in the grid is 50×50 pixels.

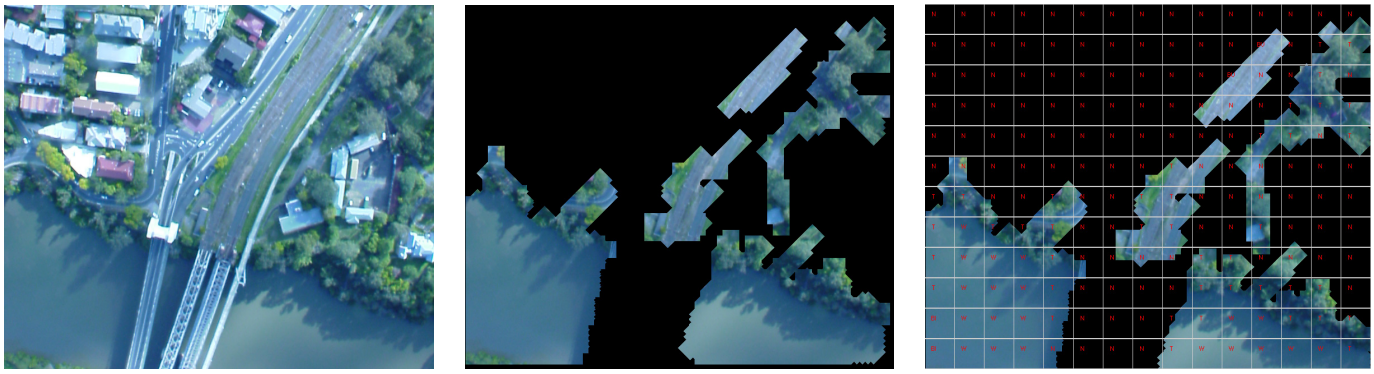

Fig. 7: Classification result on a sample image taken from data set D. Left: input image. Center: segmented image. Right: classification result over an overlaid grid. Six classes shown are BU: building, Bi=bitumen, T=trees, G=grass, W=water and N=none. Each cell in the grid is 50×50 pixels.

no. 2, pp. 121–167, June 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1009715923555

[21] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, Mar 2002.

[22] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *J. Artif. Int. Res.*, vol. 2, no. 1, pp. 263–286, Jan. 1995. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622826.1622834

[23] P. Chen and S. Liu, "An improved dag-svm for multi-class classification," in *Natural Computation, 2009. ICNC '09. Fifth International Conference on*, vol. 1, 2009, pp. 460–462.

[24] A. Gidudu, G. Hulley, and T. Marwala, "Image classification using svms: One-against-one vs one-against-all," *CoRR*, vol. 0711.2914, 2007.

[25] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning research*, vol. 5, pp. 101–141, 2004.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[27] P.-H. C. Rong-En. Fan and C.-J. Lin., "Working set selection using second order information for training svm," *Journal of Machine Learning Research*, vol. 6, pp. 1889–1918, 2005.

[28] H. Permuter, J. Francos, and I. Jermyn, "A study of gaussian mixture models of color and texture features for image classification and segmentation," *Pattern Recognition*, vol. 39, no. 4, pp. 695–706, 2006, doi: DOI: 10.1016/j.patcog.2005.10.028.

[29] D. L. Fitzgerald, "Landing site selection for uav forced landings using machine vision," Ph.D. dissertation, Queensland University of Technology, 2007. [Online]. Available: http://eprints.qut.edu.au/16510/

[30] H. J, O. V, and R. E., "Improved blur insensitivity for decorrelated local phase quantization." in *Proc. 20th International Conference on Pattern Recognition (ICPR 2010), Istanbul, Turkey*, 2010, pp. 818–821.

[31] B. Jahne, *Practical Handbook on Image Processing for Scientific and Technical Applications*. CRC Press LLC, 1997.

[32] M. P. Dubuisson-Jolly and A. Gupta, "Color and texture fusion: application to aerial image segmentation and gis updating," *Image and Vision Computing*, vol. 18, no. 10, pp. 823–832, 2000, doi: DOI: 10.1016/S0262-8856(99)00050-5.

[33] A. R. Smith, "Color gamut transform pairs," *SIGGRAPH Comput. Graph.*, vol. 12, no. 3, pp. 12–19, 1978, 807361.

[34] L. Chen, G. Lu, and D. Zhang, "Effects of different gabor filters parameters on image retrieval by texture," in *Multimedia Modelling Conference, 2004. Proceedings. 10th International*, Jan 2004, pp. 273–278.

[35] A. Hanbury, "The taming of the hue, saturation and brightness colour space," in *In Proceedings of the 7th CVWW*, 2002, pp. 234–243.

[36] H. J and O. V., "Methods for local phase quantization in blur-insensitive image analysis." in *Proc. International Workshop on Local and Non-Local Approximation in Image Processing (LNLA 2009)*, 2009, pp. 104–111.

[37] D. Dusha, L. Mejias, and R. A. Walker, "Fixed-wing attitude estimation using temporal tracking of the horizon and optical flow," *J. Field Robotics*, vol. 28, no. 3, pp. 355–372, 2011.