Unmanned Aerial Vehicle Emergency Landing Site Identification System Using Machine Vision

Sumair Aziz, Rao Muhammad Faheem, Mudassar Bashir, Adnan Khalid, and Amanullah Yasin
Center for Advanced Studies in Engineering (CASE), Islamabad, Pakistan
Email: {sumair.aziz, adnan_uet38}@yahoo.com, {rao.m.faheem, mbuetect322}@gmail.com, amanyasin@case.edu.pk

***Abstract*—Unmanned Aerial Vehicles (UAVs) rely on navigation commands from autonomous flight control system or from Ground Control System (GCS) via line-of-sight wireless data link. UAV needs to perform immediate landing on predefined airfield in case of extreme emergency like navigation, data link, engine or control surface failure, that cannot be accomplished in some cases and accidents can occur which can result collateral damage as well. This paper presents the system design which can discover the appropriate area within the surroundings for immediate landing in case of emergency. The proposed system design consists of two stages. During first stage, system takes top view images from UAV onboard camera, then image processing algorithm extracts and refine the attributes of the image. In second stage, machine learning based algorithm evaluates the results from previous stage, and based on its previous training, decides whether the area visible in image is good for safe landing or not. We implement proposed system design in MATLAB and the approach used is validated with experimental results on test data. Proposed system design uses combination of simple techniques, which makes it less computationally intensive, having reduced latency, low implementation cost and easy to implement on high speed real time hardware like FPGA/ASIC.**

***Index Terms*—vision based landing, emergency landing, machine vision, machine learning**

I. Introduction

Unmanned Aerial Systems are extensively used in commercial and military sectors e.g. security & control, aerial reconnaissance, aerial policemen and crowd monitoring, security watch, maritime search and rescue, oil and gas pipeline monitoring, disaster effects management, rescue and clear up effort supervision, disaster damage estimation, crop management, telecom relay and signal coverage survey, oil and gas exploration and geophysical surveys. UAVs accident rate is 100 times more than manned aircraft. Although UAV failure rate is about one per 2000 flight hours but analysis shows that for every 1000 flight hours there is at least one mishap [1].

UAVs mostly crash because of mechanical failures and data link loss. In case of data link loss UAVs are programmed to fly in a circular pattern until the links are restored. In worst-case scenarios, they are supposed and programmed to return autonomously to their launch base on preprogrammed route information using GPS navigation. In more than a quarter of the accidents investigation had revealed that data links were lost and UAVs had not returned to the base. Satellite/data link connections can be lost when a UAV take sharp turns or drop altitude too quickly due to failure of engine or control surfaces. Link losses can also occur in case of Data Link Station failure e.g. power loss or equipment malfunction. This link can also be disrupted due to uneven terrain and mountains between GCS and UAV due to directionality of microwaves. 45-50% of total crashes were due to engine and mechanical failure. That could be due to oil leakage, propeller malfunctioning, control surfaces (flaps, ailerons, rudder) failure [2].

In the past UAV designers had not carried out much research in improving the emergency landing site selection system. In last decade some valuable research had been carried out in the field of computer vision and machine learning to develop the emergency landing systems. The family of researchers which developed the emergency landing systems using machine learning have used Support Vector Machine (SVM) and Artificial Neural Network (ANN) algorithms along with digital image processing techniques to select the appropriate site for landing [3], [4]. Since SVM is very complex and computationally intensive technique, and ANN requires large data for training as a result it requires more training time, so both these techniques cannot satisfy the fast changing demands in emergency or forced landing area selection systems. A better and computationally less intensive approach of classification i.e. K-Nearest Neighbor (KNN) technique can be used to get the desired results. Alongside machine learning algorithms, image processing techniques like texture/features extraction are also required to get the useful information in the images data for classification purposes. Many researches have used the texture features in order to classify the images. Image data is very large, and it becomes very complex and time taking for the machine learning classifiers to train and then predict. The proposed approach used dimensionality reduction concept, in which few most important attributes are extracted from the large image data. Then extracted attributes are passed to machine learning algorithm for training and then real time test cases. Dimensionality reduction gives excellent improvement of performance in terms of processing speed and complexity.

Manuscript received April 20, 2015; revised November 2, 2015.

doi: 10.18178/joig.4.1.36-41

II. ALGORITHM OVERVIEW

This paper proposed an algorithm which combines machine learning and image processing techniques, to assist the UAV in differentiating among good and bad landing sites. In case of occurrence of emergency caused by failure of any kind as discussed above, the first step is to take top view image of the area just below UAV using its onboard camera. The attributes/feature vector of the image is computed by applying image processing techniques. The second step is to pass the computed feature vector to machine learning algorithm. In supervised machine learning techniques there are two modes, Training Mode and Test Mode which is also Real Time Mode in our case. During training stage, the image's attributes and class (bad/acceptable/good) are passed to the algorithm as training data as shown in Fig. 1. During Real Time Mode, only image's attributes are passed to algorithm and it predicts whether the area is good for landing as shown in Fig. 2.

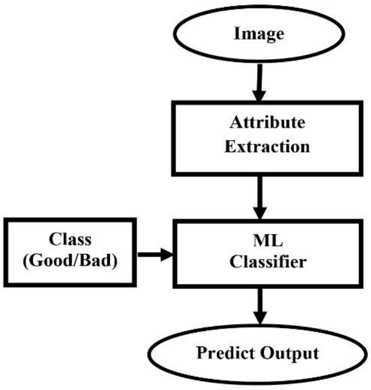


Figure 1. Training mode of algorithm

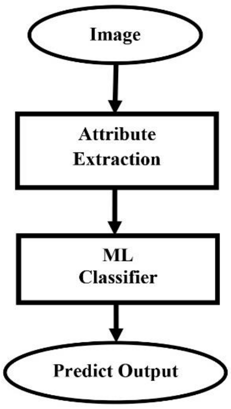


Figure 2. Real time working mode of algorithm

III. ALGORITHM PRINCIPLE

The proposed system in this paper will take top view images of the area just below UAV from the height of 1000 meters using onboard camera. Height of the UAV could be obtained from sensors like radio altimeter and GPS navigation systems [5]. The area visible from this height in one frame of 750×450 pixels is 800×500 meters. This area is large enough for landing of big size UAV [6]. Test images have been taken from Google satellite imagery shown from Fig. 3 to Fig. 5. Area visible in images is classified into three different categories.

Good Area Category consists of those images in which the areas visible are safe for landing, for example grounds, open areas, fields and water etc. as illustrated in Fig. 3.

Bad Area Category areas like mountainous terrain, forests, and houses in large number and other obstacles as shown in Fig. 4.

Acceptable Areas Category consist of the images in which almost more than 50 percent of the area visible in image is without obstacles, but there might be some houses or obstacles present as shown in Fig. 5

Figure 3. Good landing site example

Figure 4. Bad landing site example

Figure 5. Acceptable landing site example

As mentioned earlier, there are three classification categories in our context. Classification is performed by adopting a two stage process that encapsulates attribute extraction and classification. Techniques pertaining to image processing were employed for attribute extraction while supervised machine learning algorithm was used for classification. This research paper articulates good and bad examples to reflect granular processing details encapsulated in our proposed approach.

A. Attribute Extraction from Images Histogram

Attributes were extracted from the image using histogram, edge detection and masking operations as shown in Fig. 6. Among many attributes of image data we select only seven most important for classification. This dimensionality reduction gives very efficient results in terms of high speed processing, reduction in the complexity of hardware design and time to predict the class.

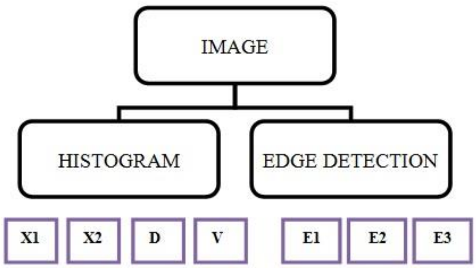


Figure 6. Attribute extraction scheme

1) Histogram threshold

For 8 bit gray scale image there could be 256 different possible intensities. So in this case histogram will show the pixel’s intensity level distribution on scale of 256 values. Consider the histograms of good and bad landing sites as discussed earlier (Fig. 3 and Fig. 4). Histogram distributions for these examples are illustrated in Fig. 7 and Fig. 8. By observing closely, most cases of the good landing sites are having narrow intensity level distribution. This means that good landing sites are mainly having same color distribution. And in most cases of bad landing sites the pixel’s intensity distribution will be wide i.e. it will be having contrast or having large color variation. This can be observed in figures shown below. This difference in the histograms of good and bad examples could be utilized by setting some threshold values along x-axis and y-axis. Four attributes have been extracted out of this histogram distribution for further processing.

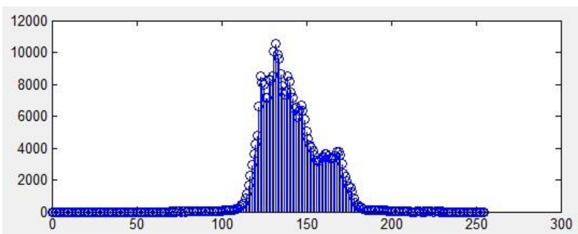


Figure 7. Histogram of good landing site

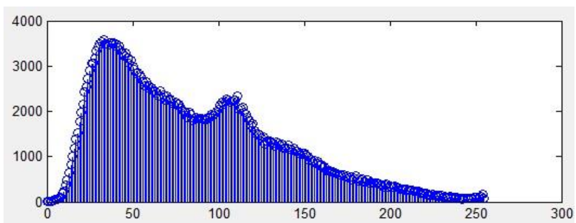


Figure 8. Histogram of bad landing site

Attributes related to histogram threshold are explained below and highlighted in Fig. 9.

X1: When the no. of pixels on certain intensity value starts exceeding 750, record that intensity value in variable X1.

X2: When the no. of pixels decrease below 750, record that intensity value in variable X2.

D: This is the measure of width of pixel’s intensity distribution.

V: This value is the peak value of the pixel’s intensity distribution. It shows that which color is having maximum number of pixels. It will be very useful in classification of water bodies, as they are mostly having same color. So this value is huge for water bodies.

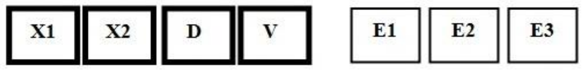


Figure 9. Attribute vector, histogram related attributes are highlighted

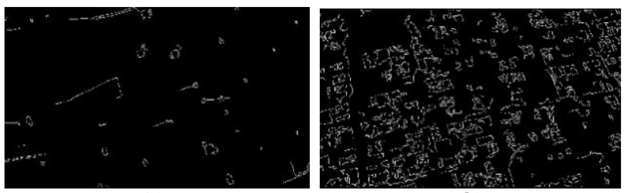
2) Canny edge detection

Canny Edge Detection algorithm is widely used edge detection operator [7], [8] and it includes the following basic steps for edge detection:

a. Low-Pass spatial frequency filtering.
b. Application of first order differential masks.
c. Non-Maximum suppression involving sub pixel interpolation of pixel intensities.
d. Hysteresis thresholding.

Canny edge detection algorithm has better performance in case of noisy conditions as compared to other edge detection operators [9]. Canny edge detection algorithm has greater accuracy in detecting object edges with high entropy, mean square ratio and peak signal to noise ratio as compared to Sobel, Roberts and Prewitt edge detection operators [10].

Different thresholds were applied but the best and optimized results were obtained at 0.45. Results of applying Canny operator on good and bad landing sites shown previously in Fig. 3 and Fig. 4, could be seen in Fig. 10(a) and Fig. 10(b).



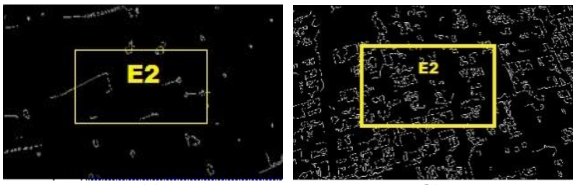
(a) (b)

Figure 10. (a) Edges computed in good lading site, (b) edges computed in bad landing site

The definition and explanation of the extracted attributes related to edge detection is given below.

E1: Counting of Canny Edges in image under processing, in case of good examples there will be less edges and vice versa.

E2: This attribute contains the counting of edges in the center of the image (rectangular box in the center of image, as the center location is very important for landing, and this value should be small, in case of good landing site, and high in case of bad landing site. E2 attribute can be seen in Fig. 11(a) and Fig. 11(b) for good and bad landing sites.



(a) (b)

Figure 11. (a) Attribute E2 for good lading site, (b) attribute E2 for bad landing site

As a result of applying edge detection technique on the image under processing; the extracted attributes are highlighted below in the attribute vector as shown in Fig. 12.

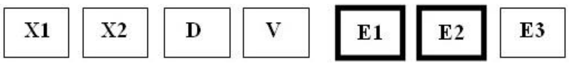
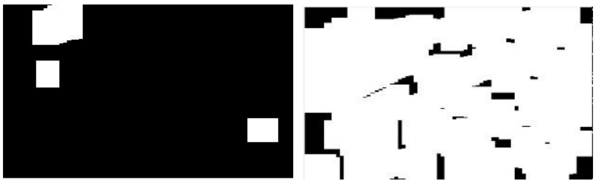


Figure 12. Attribute vector, histogram related attributes are highlighted

3) Masking operation

In image processing mask, convolution matrix or kernel is a small matrix used for performing different operations directly on the image data. For this proposed algorithm, 60×60 mask of ones is multiplied with the image data along x and y-axis. If the result of the multiplication is the matrix of zeros, it means that location in the image is having no obstacles or edges and this area is marked with white color. And if the result of multiplication contains some values, then it means that there might be edges and obstacles in that location in the image, so this area is marked with black color. The result of masking operation is shown in Fig. 13(a) and 13(b).



(a) (b)

Figure 13. (a) Results of applying masking on good landing site (b) results of applying masking on bad landing site

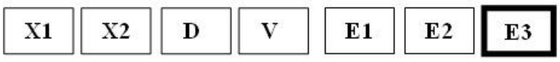


Figure 14. Attribute vector, highlighted attribute shows the value obtained as a result of masking operation

After applying masking operation on the image, attribute E3 is extracted as highlighted in Fig. 14.

B. Histogram Machine Learning Algorithm

We are using the KNN i.e. Kth nearest neighbor machine learning algorithm for classification. The basic steps of algorithm include:

- For each testing point.
- Measure distance to each training point.
- Find K closest points.
- Identify the most common class among those K.
- Predict that class.
- End.

The distance among the points is measured using the Euclidean distance formula as shown in (1).

$$d = \sqrt{(x - x_1)^2 + (y - y_1)^2} \quad (1)$$

During training mode, attribute vector needs to be labeled with the class and is made input to the KNN algorithm. This is the training mode in supervised machine learning algorithms. This is shown in Fig. 15.

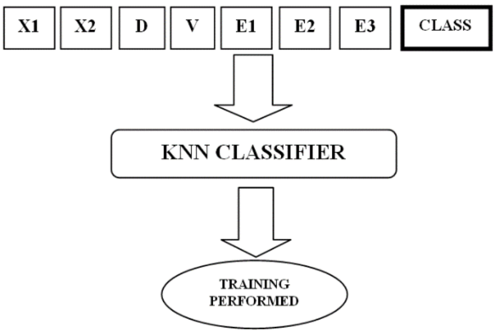


Figure 15. KNN training mode

Once the training is complete for any supervised machine learning algorithm, then it could be used to predict the output once it's given the attribute vector, which is extracted from image. During this mode, only attribute vector is given as input to the already trained machine learning algorithm. Working of algorithm in this mode is shown in Fig. 16.

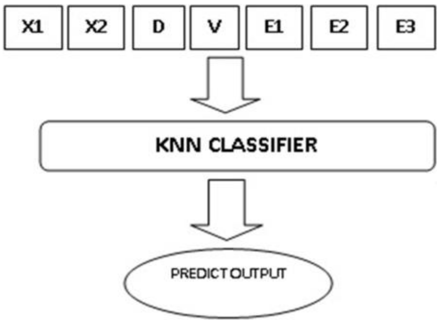


Figure 16. KNN real time mode

IV. Simulation and Results

Software and a Graphical User Interface (GUI) have been developed using MATLAB 8.3 2014a is shown in Fig. 17 and Fig. 18. Upon trying different sized masks, different threshold values for canny edge detection and different values of K for KNN machine learning algorithm, the best optimized results were obtained at and using 60×60 mask, threshold=0.45 and k=3 for KNN algorithm. Accuracy of 86.95% has been obtained and the processing time per frame is 0.034108 seconds which is very good. Software could be customized by the end user to perform in a better way, according to each mission type and terrain. This customization could be performed by little effort in the beginning by the end user, but it offers very low computational cost and high throughput. Software needs to be trained for each type of mission only first time, and training file could be saved for future for the same mission.

Due to different training data set for different type of missions, the proposed algorithm performs optimally.

The training data is only specific, related to the mission area so very small memory will be needed to store the data. Due to this unique capability, the system performs very efficiently in terms of processing time, computational complexity etc.

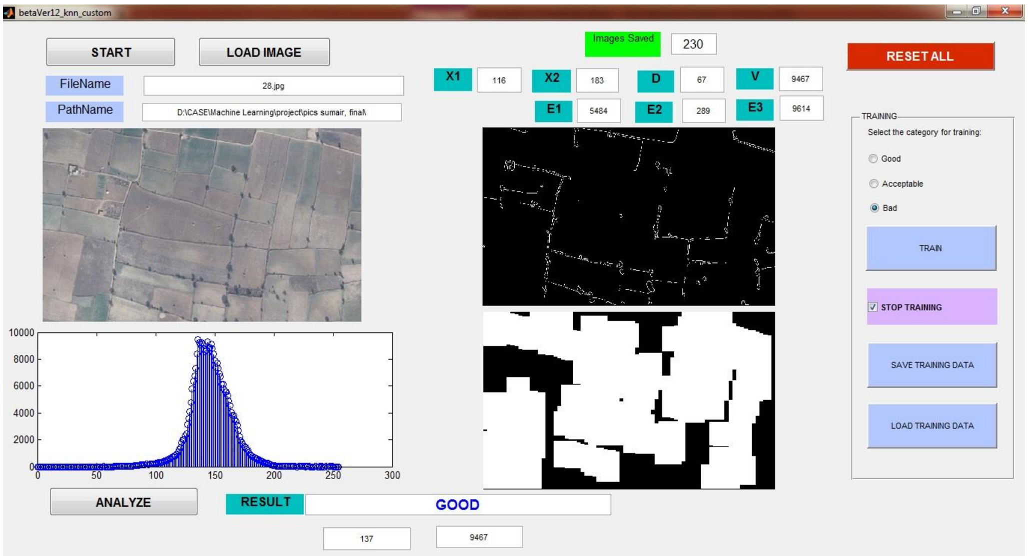


Figure 17. Software classifying good landing site successfully

Figure 18. Software classifying bad landing site successfully

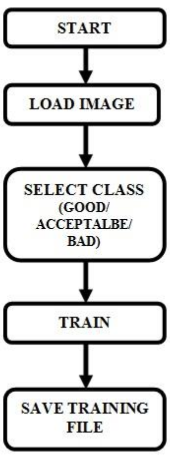


Figure 19. Software workflow in training mode

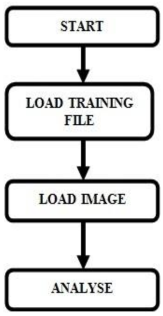


Figure 20. Software workflow in real time mode

Working of designed software for end user in training and real time mode can be seen in the Fig. 19 and Fig. 20 respectively. User can save the training file for future use in the training mode.

V. CONCLUSIONS

Commercial and military uses of UAVs have been dramatically increased in last few years. This has brought a need of full integration of UAVs in civilian airspace. Up to this time, use of UAVs is mostly limited to the areas where there is not very large population. This is due to the limitations and failures that could result in UAV crash which could result collateral damage. In this paper the researchers have pointed out the solution to the problem in shape of finding safe area for forced or emergency landing. This is the first important step in performing the forced landing autonomously. The proposed algorithms and techniques used in this paper achieved 86% accuracy upon training on small data set that is of 230 images, which is acceptable. The actual requirement of images for a good landing execution depends on the type of terrains for which user wants to train the algorithm. However in general scenarios about 200 images are quite enough for a good landing execution. KNN machine learning technique has been used to keep the algorithm simple for implementation on real time hardware like Field Programmable Gate Arrays (FPGA). Processing time on each frame took only 0.034108 seconds which is very good in terms of throughput. Proposed algorithms in this paper are ready to be implemented on FPGA devices. It requires the hardware/software partitioning and conversion of the floating point architectures on fixed point hardware. It will further reduce the processing time, hardware cost and power requirements of the real time system.

REFERENCES

[1] C. M. David, W. King, and A. Bertapelle, "UAV failure rate criteria for equivalent level of safety," in *Proc. International Safety Symposium*, Montréal, Québec, Canada, 2005.

[2] C. Witlock. (2014). When drones fall from the sky. The Washington Post, Washington DC, USA. [Online]. Available: http://www.washingtonpost.com/sf/investigative/2014/06/20/when-drones-fall-from-the-sky/

[3] D. Q. Zhao, W. W. Zou, and G. M. Sun, "A fast image classification algorithm using support vector machine," in *Proc. International Conference on Computer Technology and Development*, 2010.

[4] S. M. S. Rashmi and S. Mandar, "Textural feature based image classification using artificial neural network," *Advances in Computing, Communication and Control*, vol. 125, pp. 62-69, 2011.

[5] A. Y. Lu, W. R. Ding, and H. G. Li, "Multi-information based safe area step selection algorithm for UAV emergency forced landing," *Journal of Software*, vol. 8, no. 4, pp. 995-1002, April 2013.

[6] D. R. W. D. Fitzgerald, R. Walker, and D. Campbell, "A vision based emergency forced landing system for an autonomous UAV," in *Proc. Australian International Aerospace Congress Conference*, Melbourne, Australia, 2005.

[7] X. M. Li, "A software scheme for UAV's safe landing area discovery," in *Proc. AASRI Conference on Intelligent Systems and Control*, Vancouver, Canada, 2013.

[8] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[9] D. C. G. T. Shrivakshan, "A comparison of various edge detection techniques used in image processing," *International Journal of Computer Science Issues*, vol. 9, 2012.

[10] P. P. Acharjya and R. Das, "Study and comparison of different edge detectors for image segmentation," *Global Journal of Computer Science and Technology Graphics & Vision*, 2012.

Sumair Aziz received bachelor degree in Electronic Engineering from International Islamic University, Islamabad in 2012. He has research and development experience of more than two years. His research interest includes the development of vision based intelligent systems, embedded systems and robotics.
His awards and honors include the 1st Position in Computer Logic Design Competition NASCON-2011 FAST-NU Islamabad, 1st Position in Robotics Competition (Fire Fighting) DICE-2011 UET Peshawar, Won "Best External Project" Award at ROBIAN–2011 COMSATS Robotics Competition, Islamabad, 1st Position in Mico-Maze Competition at Students Engineering Exhibition PAF-KIET Karachi and 2nd Position in FYP Project Competition at 3rd Open Hoise - 2012 International Islamic University Islamabad.

Rao Muhammad Faheem received bachelor degree in Electrical (Telecommunication) Engineering from FAST-National University of Computer and Emerging Sciences, Islamabad in 2012. Now he is going to receive M.S. degree from Center for Advanced Studies in Engineering (CASE), University of Engineering and Technology. He has three years of experience in Research and Development. His research interest includes artificial intelligence, machine vision, image processing and signal processing.

Mudassar Bashir received bachelor in Electronic Engineering from University of Engineering and Technology, Taxila in 2012. He has more than two years of experience in Research and Development. His research interests are navigation systems, RF electronics and signal processing.

Adnan Khalid is doing Masters in Electrical Engineering from Centre for Advanced Studies in Engineering Sciences, Islamabad. He received his Telecommunication Engineering Degree from University of Engineering and Technology, Taxila, Pakistan in 2011. His research interest includes Digital Signal Processing, Antenna and Machine Learning.

Dr. Amanullah Yasin is an Assistant Professor in Center for Advance Studies in Engineering (CASE) Islamabad Pakistan. He received his PhD degree in Data Mining, School of engineering, University of Nantes, France and MS degree in Knowledge Engineering from University of Nantes, France. His research interest including data mining, data stream mining, machine learning, knowledge engineering and Business intelligence.