Wireless Video Noise Classification for Micro Air Vehicles

Jeffrey Byrne and Raman Mehra
Scientific Systems Company, Inc.
+1 781 933 5355, +1 781 938 4752 (fax)
{jbyrne,rkm}@ssci.com

Abstract

Onboard processing of video is currently outside the capabilities of power limited micro air vehicles (MAVs), which forces researchers to transmit video and telemetry to a ground control station for capture and offline processing. Unfortunately, wireless video transmission can introduce structured noise, which can corrupt image processing algorithms if the noisy frames are not identified and rejected from further processing. In this paper, we describe the design and evaluation of a supervised learning based classifier for labeling frames of video "noisy" or "clean". We pose the classification problem as one of texture classification in video, where texture is represented using a feature set including informative statistics of steerable pyramid coefficients and the principal components of the chromatic histogram. The main contribution of this paper is the definition of this feature set as determined from analysis of noisy video. We evaluate nine different binary classifiers using cross validation on a MAV video training set, with additional evaluation on two validation sets collected from different times, days and locations. Results show that the best performing classifier is stepwise logistic regression, with a cross-validation accuracy of 0.89.

1 Introduction

Micro Air Vehicles (MAVs) are small, lightweight, and autonomous aerial systems that can fit in a backpack, and promise to enable on-demand intelligence, surveillance and reconnaissance tasks in a near-earth environment. Such tasks for military operations may include: “over the hill” reconnaissance, “perch and stare” surveillance, covert imaging, biological and chemical agent detection, tagging and targeting, precision strike missions and bomb impact indication. Civil and commercial applications for MAVs are not as well developed, although potential applications are extremely broad in scope. Possible applications for MAV technology include: first responder intelligence gathering for CBRNE (chemical, biological, radiological, nuclear, explosive) events, environmental monitoring (e.g., pollution, weather, and scientific applications), forest fire surveillance, border patrol, drug interdiction, aerial surveillance and mapping, traffic monitoring, precision agriculture, disaster relief, ad-hoc communications networks, and rural search and rescue.

Research and development for MAV design has mainly focused on the airframe aerodynamics [1], as well as guidance, navigation and control [2] using onboard miniaturized avionics. Recently, researchers have explored processing of onboard color video for active control [3], obstacle detection [4], structure from motion [5] and color segmentation [6]. However, MAVs at the size of future combat systems (FCS) class I UAVs or smaller have significant power, size and weight constraints which limit the onboard processing of sensor information. Real time video processing using modern computer vision algorithms is within the scope of larger UAVs and ground vehicle who can process live video or collect and store video onboard, but is currently outside the scope of onboard MAV processing due to power constraints.

A common solution to this problem is to include a wireless video transmitter on the MAV, then transmit video and telemetry to the ground for offline analysis or ground-station-in-the-loop control. Color NTSC video on the MAV is modulated onto a carrier in the unlicensed ISM bands (900MHz, 2.4GHz), then demodulated at the ground control station for digitization. Interference from other unlicensed transmitters (802.11, bluetooth, garage door openers, etc.), analog noise in the embedded avionics, and multi-path effects can corrupt the video signal during transmission, introducing distortion in the video received at the ground. Unlike a larger UAV, a MAV cannot increase the transmit power due to power constraints, and digital transceivers have not yet been miniaturized for integration into MAV avionics, so the video must be compensated for at

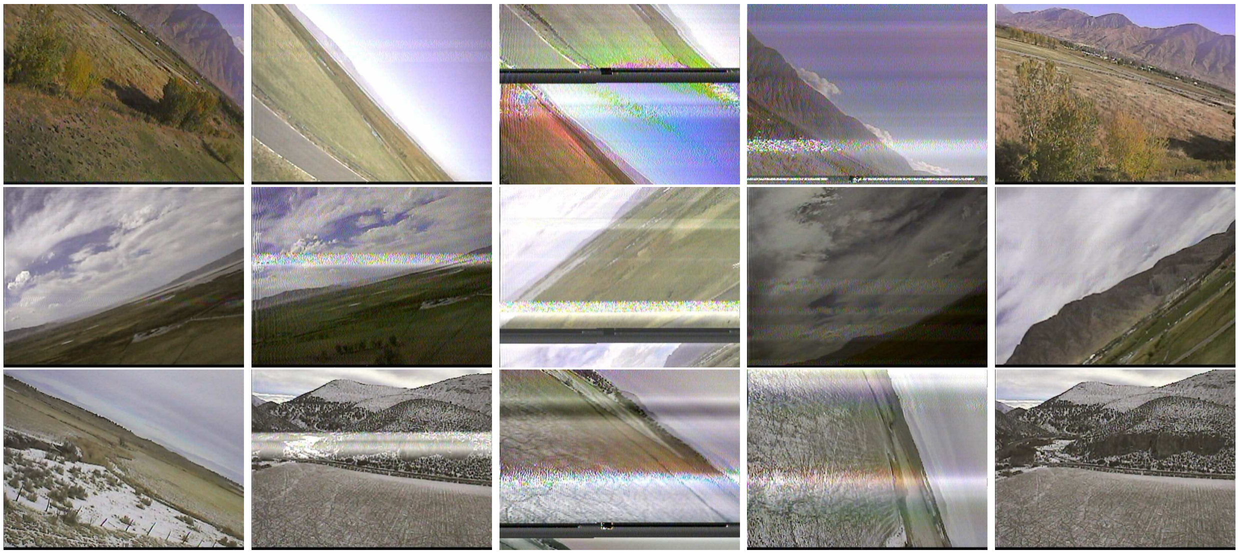


Figure 1: Wireless video noise examples. Each row is a sampling of imagery from a given dataset, where the first and last columns show clean imagery, the left middle and right middle columns show moderately noisy imagery, and the middle column shows severely noisy imagery. The noisy imagery shows chromatic noise, horizontal scanline noise and visible vertical blanking due to vertical sync loss.

the receiver. NTSC analog noise filters at the receiver can help with chromatic distortion, and digital noise filtering techniques can help with independent, identically distributed (IID) noise, however neither will work for *structured noise*, as is introduced with an strong unlicensed transmitter nearby, or loss of horizontal or vertical sync. Such noise from the wireless transmission will impact any video processing algorithm, so therefore, it would be beneficial to detect when this noise is present so that a noisy frame can be rejected from further processing. Furthermore, since the properties of the noise are dependent on the transceiver, platform and environmental properties, it is reasonable to propose a classification algorithm that can be trained based on labeled examples of noisy frames, so that the classifier can be adapted to the current conditions. Such an approach to *supervised wireless video noise classification* would aid the researcher in the test and evaluation of vision processing for video data requiring wireless transmission.

Figure 1 shows examples of wireless video noise captured from a small unmanned air vehicle. Each row is a sampling of imagery from a dataset collected at various times and under various flight conditions. Row one shows nominal flight, row two shows video collected at the same location as nominal flight, but on a different day and time with aggressive maneuvers. Notice the overcast background, but similar terrain. Row three shows video collected from a different location on a different day and time, which includes snow on the ground and mountainous terrain. The first and last columns show clean frames of video without any noise, and the middle three columns show moderate to severe noise including: chromatic noise, scanline noise and blanking noise. *Chromatic noise* is present when the video color subcarrier is distorted at the receiver, which manifests as atypical colors in the scene. *Scanline noise* is high frequency noise that corrupts sets of scanlines, leaving the remainder of the video clean. Finally, *blanking noise* shows a black bar, which is the vertical blanking interval of the NTSC signal, which appears when the receiver loses vertical sync on the video and the video exhibits the characteristic ”rolling” motion until the vertical sync is reacquired.

In this paper, we describe the design and evaluation of a supervised learning based classifier for labeling frames of video ”noisy” or ”clean”. This classifier can be applied as a preprocessing step to reject noisy frames from further processing, so that computer vision algorithms operate only on imagery with contrast that is consistent with the scene, and not with the transmission artifacts. We pose this problem as a texture classification problem, where texture is represented in terms features including statistics of the steerable pyramid and principal components of the chromatic histogram. The main contribution of this paper is the definition of the feature representation suitable for video noise classification as described in section 2. Section 2.2 describes the classifiers evaluated and the training methodology and benefits for each. Finally, we collected three labeled datasets containing a total of 350 hand labeled noisy and clean image frames extracted from MAV video from different times, days and locations to show performance of the classifier on new video. Section 3 shows the performance results of each classifier on 10-fold cross validation on the training set, as well as the ROC curves and area under ROCs for performance on the validation set.

2 Technical Approach

We pose the wireless video noise classification problem as one of *texture classification* given chromatic, scanline and blanking noise. Figure 1 show examples of wireless video noise where we observe that noise is often localized to a set of contiguous scanlines of various widths, due to the period in which an unlicensed transmitter is transmitting and the shutter timing of CMOS cameras. This noise can be described as a *stochastic texture* since within the contiguous scanlines, the noise appears drawn from a common, but unknown distribution. Supervised texture classification is the process of training a classifier to detect the presence or absence of a given stochastic texture pattern in an image.

Texture classification requires a representation of texture. Observe that noisy frames in figure 1 have stochastic texture that is often limited to a minimum of approximately 11 scanlines, which we will call scanline bundles or simply *bundles*. Also, observe that this noise is not a uniform random distribution of RGB colors within a bundle, rather there is a dominant orientation which can be measured using a scaled and oriented filter bank. One such filter bank, the *steerable pyramid* [7, 8], is an efficient and accurate linear decomposition of an image into scale and orientation sub-bands. The basis functions of this decomposition are scaled and oriented kernels which are rotated copies of one another, where such steerable kernels can be used to reproduce directional bands of any order. The steerable pyramid improves on orthonormal wavelet decomposition by eliminating aliasing in reconstruction, and introducing a steerable orientation decomposition such that any orientation response can be determined as a linear combination of the basis. Intuitively, wireless transmission noise will introduces coefficient responses at scales and orientations with different magnitudes than natural features. We will exploit these features for classification.

Chromatic noise can be represented using principal components features of the chrominance histogram. An RGB image can be converted into YUV color space, and a joint histogram for the U and V components can be computed. From this, the N principal components are determined from training and the loadings are used for additional features.

Blanking noise can be represented using the phase alignment of multiscale horizontal steerable pyramid coefficients along a scanline. In an NTSC signal, the vertical blanking interval is the period of time between the end of a field and the start of the next field, where non-visible information such as synchronization and metadata may be stored. However, when an NTSC receiver loses vertical synchronization due to interference, the vertical blanking portion of the signal becomes visible and manifests as a horizontal black bar. This black bar can be detected using steerable pyramid bands with orientation tuned to horizontal edges ($\theta = 90^o$), and observing that the phase of the filter responses over all scales will be aligned on a single scanline. Maxima of this phase alignment is an indication of a visible vertical blanking interval.

2.1 Feature Set

The feature vector used for this investigation was chosen to reflect the three types of distortion discussed in section 1. The chromatic noise feature is computed over the entire image to model the color distortion which affects the whole scene in some noise scenarios. The bundle features are computed over all bundles (e.g. subset of contiguous scanlines) in the image to model local scanline distortion. The bundle width was chosen from observation to be 16 scanlines. For each bundle, we compute the blanking feature and the orientation energy (equation 1) at scale 1 and orientation 0^o, then select the bundle with the maximum response. Intuitively, the bundle a strong blanking feature has phase aligned horizontal edges along scanlines and a bundle with strong OE has strong yet small scale vertical edges, both are consistent with observations.

Given this selected bundle, we compute the remaining bundle statistics, carefully chosen to reflect a property of the underlying noise, enabling separability for the classifier. The result is a 43x1 feature vector for each image to be used in classification. For each image, we convert the RGB input image into YUV color space, then compute the chrominance (UV) joint histogram. Next, we compute four bands of the steerable pyramid decomposition for the luminance (Y) component, for two orientations $(0^o, 90^o)$ and two scales. Finally, we compute the following statistics for the feature vector, with the associated lengths.

1. **Principal components of the UV histogram (16):** We compute the 16 principal compo-

nents of a histogram computed from the chrominance components (UV) of a YUV color image. The histogram is quantized to 8x8 and is computed for the entire image. Features are the projected coefficients into the principal components. This feature is included to model atypical color introduced by color subcarrier distortion.

2. **Bundle mean (4):** Compute the mean of each steerable pyramid band for the selected bundle (see above for bundle selection). Noise bundles are expected to have strong edge response at the smallest scale due to the random appearance.
3. **Bundle variance (4):** Compute the variance of each steerable pyramid band for the selected bundle. Noise bundles are expected to have large variance across all orientations and scales.
4. **Bundle orientation energy (4):** the orientation energy is the sum of squared coefficient responses at each orientation for the selected bundle. Given a band B_{ij}, with orientation i and scale j, the orientation energy for the selected bundle is defined as:

   $$\theta_{ij} = E(B_{ij}(x)^2) \tag{1}$$

   where E is the expectation over all coefficients in the band. Noise bundles are expected to have strong orientation energy at the smallest scale.
5. **Bundle cross-correlation (6):** The cross correlation is a measure of the independence of two bands. Intuitively, a natural scene should exhibit strong cross correlation between bands due to phase alignment and cornerness, while a noise scene should not exhibit as strong a correlation. Given the bands B_{ij}, the band cross correlation is defined as:

   $$\rho_{X,Y} = \frac{Cov(B_X, B_Y)}{\sigma_X \sigma_Y} \tag{2}$$

   Cross correlation is computed for all bands $X = (i, j)$ such that $X \neq Y$.
6. **Bundle peak signal to noise ratio (4):** The peak signal to noise ratio is a measure of the ratio of the signal to the background noise for a given band. Assuming that the signal can be characterized as the maximum coefficient response within a band, then the peak signal to

noise ratio for the selected bundle is defined as:

$$PSNR_{i,j} = 10log_{10}\frac{max(B_{ij})}{\sigma_{ij}} \tag{3}$$

Noise bundles should exhibit small peak signal to noise ratios relative to a natural scene.

7. **Bundle entropy (4):** The bundle entropy is a measure of the uniformity of the distribution of the band coefficients within a bundle. From observations of the bundle noise, one would expect the noise to be uniformly distributed within a bundle. Given a spatial histogram with N=16 bins within a bundle as an estimate for p(x), define the entropy $e_{ij} = -\Sigma_x p(x) log_2 p(x)$.
8. **Blanking distortion feature (1):** The vertical blanking distortion feature is

$$vb^* = max_s \left[E(B(s)^2_{(90^o,1)}) + E(B(s)^2_{(90^o,2)})\right] \tag{4}$$

computed as the maximum over all scanlines s. Images exhibiting blanking noise due to the appearance of the vertical blanking period during sync loss will have a scanline aligned horizontal edge that is phase aligned across scales.

2.2 Classification Algorithms

Given a feature set defined by the feature vector in section 2.1, and a set of labels {'Noisy','Clean'} defined by an operator, we can pose the problem of wireless video noise classification as a *supervised learning* problem. We evaluate the following classification algorithms [9]:

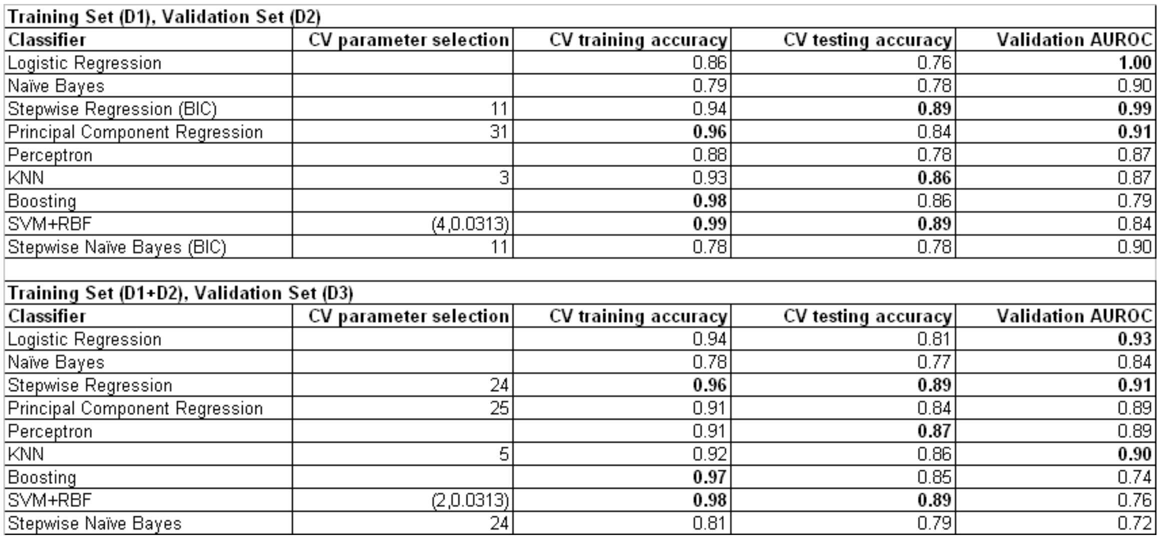
1. **Stepwise logistic regression:** We use logistic regression classifier with and without a stepwise feature selection with Bayesian Information Criterion (BIC) error term to determine the relative performance of features in the set, followed by a retrain of the logistic regression classifier using selected features.
2. **Stepwise Naive Bayes:** We use a naive Bayes classifier with and without a stepwise feature selection with a BIC error term to determine the relative performance of features in the set, followed by a retrain of the Naive Bayes classifier using selected features.

3. **Principal component logistic regression:** The number of principal components is chosen using cross-validation.
4. **K-Nearest Neighbors (KNN):** The parameter K for the number of nearest neighbors for comparison is chosen using cross-validation.
5. **Perceptron:** We include an online method here to see if in the future, the classifier can be applied online on the vehicle to adaptively react to changing noise conditions. We use 10 passes over the data.
6. **Boosted decision trees**: Decision tree classifier is trained using an adaboost procedure.
7. **Support Vector Machine (SVM):** A support vector machine with a radial basis function kernel (SVM-RBF). The error penalty C and the RBF parameter γ are chosen using cross-validation.

The stepwise classifiers include a feature selection method to determine those features in the feature set that best improve the cross validation performance. This is a means to improve overall classification performance, as well as evaluating which features from section 2.1 are informative for classification.

2.3 Data collection

Video for this investigation was collected on a fixed wing small UAV platform by Brigham Young University. BYU's UAV is a fixed wing, hand launched UAV with a 1.5m wingspan, constructed with an EPP foam core covered with Kevlar. This design was selected for its durability, usable payload, ease of component installation, and flight characteristics. The airframe can carry a 0.4kg payload and can remain in flight for over 45minutes at a time. Additional payload includes the Kestrel autopilot, batteries, a 1000mW, 900MHz radio modem, a 12-channel GPS receiver, and a video transmitter. On the ground control station, a communication box contains a 900MHz transceiver, a GPS unit, and an interface to an RC transmitter which can be used to maneuver the airframe manually. In addition to standard telemetry, we also connect the video feed from the cameras to an Imperx VCE-PRO PCMCIA frame grabber hosted on the laptop. The frame grabber



Training Set (D1), Validation Set (D2)				
Classifier	**CV parameter selection**	**CV training accuracy**	**CV testing accuracy**	**Validation AUROC**
Logistic Regression		0.86	0.76	**1.00**
Naïve Bayes		0.79	0.78	0.90
Stepwise Regression (BIC)	11	0.94	**0.89**	**0.99**
Principal Component Regression	31	**0.96**	0.84	**0.91**
Perceptron		0.88	0.78	0.87
KNN	3	0.93	**0.86**	0.87
Boosting		**0.98**	0.86	0.79
SVM+RBF	(4,0.0313)	**0.99**	**0.89**	0.84
Stepwise Naïve Bayes (BIC)	11	0.78	0.78	0.90

Training Set (D1+D2), Validation Set (D3)				
Classifier	**CV parameter selection**	**CV training accuracy**	**CV testing accuracy**	**Validation AUROC**
Logistic Regression		0.94	0.81	**0.93**
Naïve Bayes		0.78	0.77	0.84
Stepwise Regression	24	**0.96**	**0.89**	**0.91**
Principal Component Regression	25	0.91	0.84	0.89
Perceptron		0.91	**0.87**	0.89
KNN	5	0.92	0.86	**0.90**
Boosting		**0.97**	0.85	0.74
SVM+RBF	(2,0.0313)	**0.98**	**0.89**	0.76
Stepwise Naïve Bayes	24	0.81	0.79	0.72

Figure 2: Classification Results using 10-fold cross validation

provides 640X480 RGB images at 30Hz. The image can be displayed in the virtual cockpit ground control station software and processed for image in the loop applications. For this application, the UAV was flown manually with aggressive maneuvers to introduce wireless video noise.

Data collection involved recording video in three datasets: nominal (D1), aggressive (D2), and new environment (D3). The nominal flight was over rural terrain with low noise, the aggressive flight was over the same rural terrain with overcast conditions and aggressive maneuvers to increase the noise, and the final flight was over a new environment. The goal of the data collections were to gather video of new scenarios to see how the classifier scales to new data, such as same environment but different day, or new environment but same vehicle. We exported 350 video frames from these videos as images, and manually labeled each frame as 'Noisy' or 'Clean'. Examples from dataset D1 are shown in figure 1(row 1), examples from dataset D2 are shown in figure 1(row 2) and examples from D3 are shown in figure 1(row 3). These frames were processed to extract the feature vector from section 2.1 and concatenated into a labeled dataset for classification.

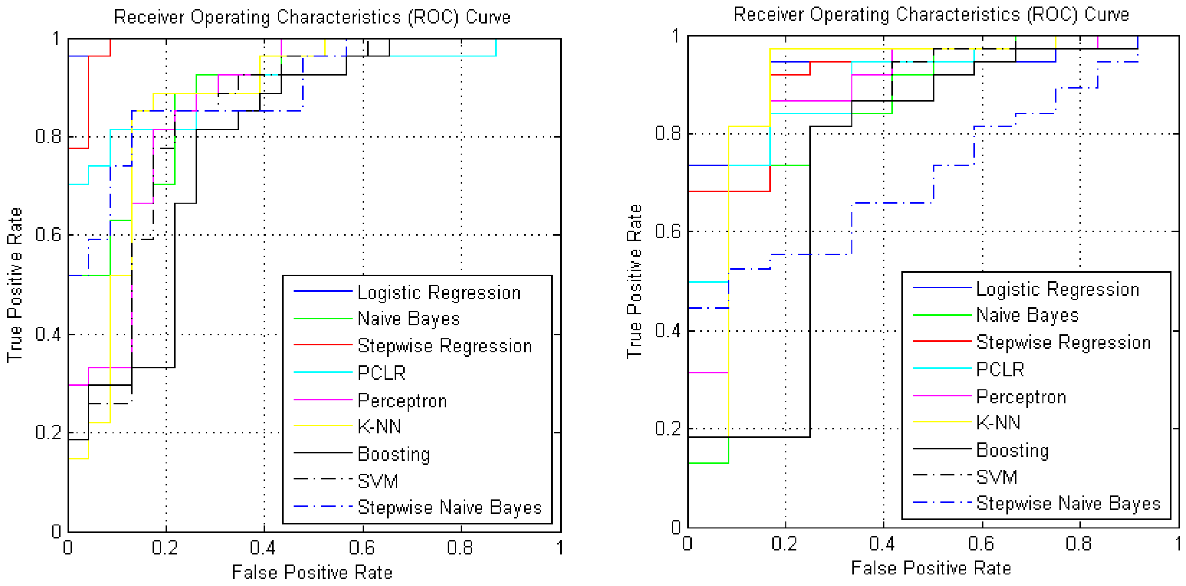


Figure 3: ROC curve for classification performance on validation set. (left) Validation on video taken over the same environment on a new day (Training D1, Validation D2). (right) Validation on video taken over a new environment on a new day. (Training D1+D2, Validation D3)

3 Experimental Results

Figure 2 shows classification accuracy results for each algorithm given 10-fold cross validation. Results are shown as classification accuracy, which is the percentage of correct classifications, both in sample (training error) and out of sample (testing error) for 10-fold cross-validation. As discussed in section 2.3, we vary the training set based on the data collection to determine the relative performance of the noise classification if the time of day is changed, or the environment is changed. Figure 2 (top) shows the cross-validation results given training data from one flight over one area (D1) with a separate validation set (D2) over the same area, but a different time of day. Figure 2(bottom) shows the cross-validation results given training data from two flights over the same area, at different times of day and different environmental conditions (D1+D2), and a validation set from a new environment (D3). In each case, the classifier is trained using cross-validation on the training set, and then the resulting model is tested on the validation set. The top three performing algorithms are shown in boldface. Figure 3 shows the receiver operating characteristics (ROC) curve for the validation set tests.

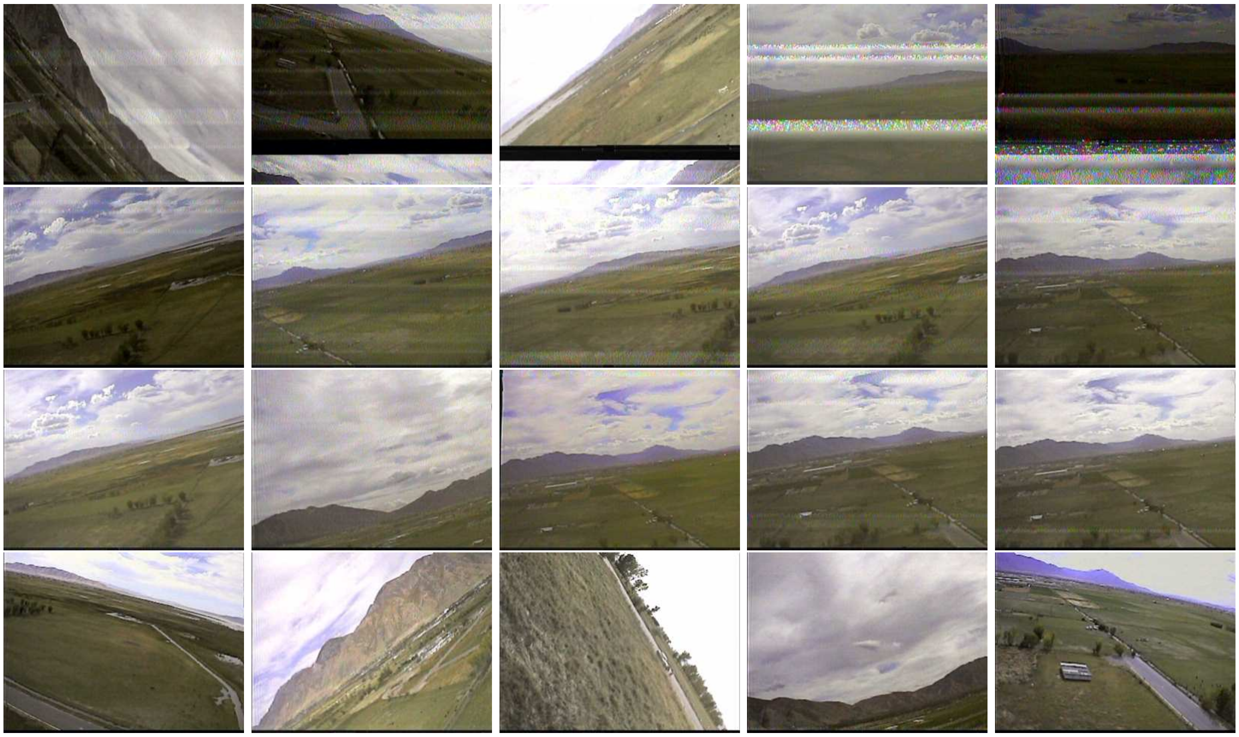


Figure 4: Wireless video classification examples. Each row shows five examples of a classification result within a given prediction range. (Row 1) Noisy: 0.95-1.0 (Row 2) Slightly noisy: 0.5-0.95 (Row 3) Clean: 0.05-0.5 (Row 4) Very clean: 0-0.05. These results show that the classifier predictions qualitatively match human intuition about ”noisy” and ”clean” as the noise gets lower and the prediction approaches zero.

Figure 4 shows examples of successful classification using stepwise logistic regression. Each classifier provides a real valued prediction p in the range ($0 \leq p \leq 1$) such that $p = 1$ corresponds to noisy and $p = 0$ clean. The noisy examples ($0.95 \leq p \leq 1$) include the extreme noise due to vertical sync loss and heavy distortion. The slightly noisy examples ($0.5 \leq p \leq 0.95$) have noise, but it is not as strongly evident as with the noisy predictions, and do not exhibit vertical sync loss. The clean examples ($0.05 \leq p \leq 0.5$) have very slight noise that is almost unnoticeable to a human viewer and requires zooming in on the image in the PDF to see. The very clean examples ($0 \leq p \leq 0.05$) have no noise at all. Observe that the prediction performance closely matches what a human would define as ”noisy” and ”clean”.

4 Analysis and Conclusions

The results in section 3 show that the most promising algorithm tested is stepwise logistic regression an average cross-validation accuracy of 89%, with an average area under ROC of 0.95. The largest factors that affected performance in our experiments was the choice of feature set that best captured the noise present in an image, and the creation of the dataset for training. We experimented with a number of different stochastic texture representations including color entropy, color co-occurrence, co-occurrence statistics, multiple scales, orientation histograms using the steerable pyramid, and higher band moments before arriving at the feature set described in 2.1. The second factor that affected the performance was the structure of the dataset. Initially, we created an unbalanced dataset with 400 clean images and 100 noisy images, where only a few of the noisy images were the difficult weak noise cases. We found that by increasing the number of noisy images, and introducing additional weak noise cases (e.g. figure 4, rows 2,3), we were able to improve the overall classification performance.

The stepwise feature selection provides a means for determining those features in section 2.1 that are most informative for classification. Stepwise logistic regression chose 11 features out of 43, where features (1,5,18,31,35,39,42,43) were consistently chosen across all cross validation folds, implying that these are good features for separability. These features are two principal components of chromatic histogram (1,5), bundle mean for small scale horizontal features (18), bundle cross correlation between scales (31), bundle peak signal to noise ratio for vertical features (35), bundle entropy for small scale vertical and large scale horizontal features (39,42), vertical blanking feature (43). The results for stepwise logistic regression in figure 4 were generated using only these features during training.

Those images that are misclassified often include natural features that mimic structured noise. For example, figure 5 shows five misclassified images ($p \geq 0.5$). From left to right in figure 5 (i) Predicted noisy, but clean. Notice that the rocks and snow on the ground have a high frequency periodic texture that is similar in appearance to the scanline noise. (ii) Predicted clean, but noisy. Notice that the selected bundle does pick up on the scanline noise in the sky, however this noise

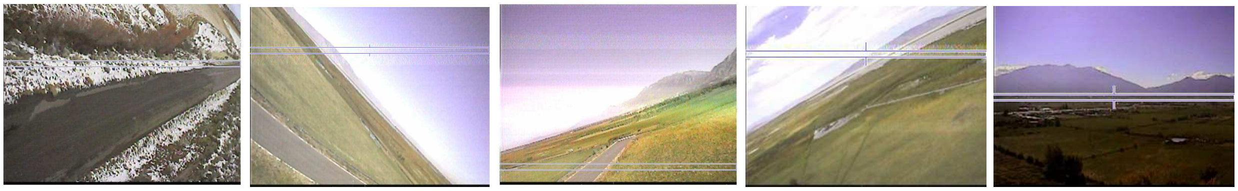


Figure 5: Misclassification examples. For each, the selected bundle used for feature extraction is shown in the image. In each case, misclassifications occur when natural features in the scene exhibit locally similar appearance to structured noise.

has texture characteristics weaker than natural textures in the scene, so is incorrectly classified (iii) Predicted clean, but noisy. Notice that the scanline features a human would pick up on in the sky are weak, so the algorithm instead picks up on strong high frequency noise in the grass texture. (iv) Predicted clean but noisy. The noise in the sky is partially washed out due to saturation effects, and all training examples assume that noise manifests across an entire scanline. (v) Predicted noisy but clean. The horizon line below the mountains exhibits strong contrast that happens to be aligned with a scanline due to the current camera orientation, which the classifier interprets as vertical blanking or scanline aligned noise. For this example, the orientation of the aircraft was just right such that it introduced horizontal features into the image along an entire scanline, which is not common. However, this case may be common with other environments and must be considered in more detail. In the false negative cases, the scanline noise is very weak, and while a human may be able to detect it, the trained classifiers incorrectly classify such images as clean. Other difficult cases include the MAV pointing the camera into the sun, introducing a chromatic distortion that was labeled noisy, or an image with partial weak chromatic distortion that does not exhibit strong response in the chromatic histogram. The remaining classification errors after considering the above cases are all borderline where often a human would have a difficult time assigning a clear label of noisy of clean as demonstrated in rows two and three of figure 4.

The analysis of the classification and misclassification results shows that severe analog noise is successfully classified, however there are conditions in which natural features have similar local texture statistics to weak analog transmission noise introducing missed detections and false positives. This implies that global image context is needed to further reduce classification errors.

References

[1] R.J. Wood et. al, "An autonomous palm-sized gliding micro air vehicle: Design, fabrication, and results of a fully integrated centimeter-scale mav," *IEEE Robotics and Automation Magazine*, vol. 4, no. 2, pp. 82–91, June 2007.

[2] Randal Beard et. al, "Autonomous vehicle technologies for small fixed wing uavs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 1, pp. 92–108, January 2005.

[3] Paul Y. Oh William E. Green and Geoffrey Barrows, "Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, May 2004, vol. 3, pp. 2347–2352.

[4] J. Byrne, M. Cosgrove, and R. Mehra, "Stereo based obstacle detection for an unmanned air vehicle," in *IEEE International Conference on Robotics and Automation (ICRA'06)*, May 2006.

[5] Omead Amidi Takeo Kanade and Qifa Ke, "Real-time and 3d vision for autonomous small and micro air vehicles," in *IEEE Conf. on Decision and Control (CDC 2004)*, December 2004, pp. 1655–1662.

[6] Randal W. Beard Joshua Redding, Timothy W. McLain and Clark Taylor, "Vision-based target localization from a fixed-wing miniature air vehicle," in *American Control Conference*, Minneapolis, Minnesota, June 2006.

[7] E.P. Simoncelli and W.T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proceedings of the Second International Conference on Image Processing*, 1995.

[8] W T Freeman and E H Adelson, "The design and use of steerable filters," *IEEE trans. on Pattern Analysis and Machine Intelligence*, 1991.

[9] Robert Tibshirani Trevor Hastie and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, 2001.