10-2014

Machine Learning Nuclear Detonation Features

Daniel T. Schmitt

Gilbert L. Peterson
Air Force Institute of Technology

Recommended Citation

Schmitt, D. T., & Peterson, G. L. (2014). Machine learning nuclear detonation features. 2014 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 1–7. https://doi.org/10.1109/AIPR.2014.7041936

MACHINE LEARNING NUCLEAR DETONATION FEATURES

Daniel T. Schmitt, Gilbert L. Peterson[*]

The Air Force Institute of Technology
Dept. of Elec. & Comp. Eng.
WPAFB, OH 45433
daniel.schmitt@afit.edu

ABSTRACT

Nuclear explosion yield estimation equations based on a 3D model of the explosion volume will have a lower uncertainty than radius based estimation. To accurately collect data for a volume model of atmospheric explosions requires building a 3D representation from 2D images. The majority of 3D reconstruction algorithms use the SIFT (scale-invariant feature transform) feature detection algorithm which works best on feature-rich objects with continuous angular collections. These assumptions are different from the archive of nuclear explosions that have only 3 points of view. This paper reduces 300 dimensions derived from an image based on Fourier analysis and five edge detection algorithms to a manageable number to detect hotspots that may be used to correlate videos of different viewpoints for 3D reconstruction. Furthermore, experiments test whether histogram equalization improves detection of these features using four kernel sizes passed over these features. Dimension reduction using principal components analysis (PCA), forward subset selection, ReliefF, and FCBF (Fast Correlation-Based Filter) are combined with a Mahalanobis distance classifiers to find the best combination of dimensions, kernel size, and filtering to detect the hotspots. Results indicate that hotspots can be detected with hit rates of 90% and false alarms ¡ 1%.

Index Terms— Dimensionality Reduction, Feature Detection, Classification

1. INTRODUCTION

During the 1950s and 1960s, the United States conducted and filmed over 200 atmospheric nuclear tests. In the past few years, for preservation purposes, these films are being digitized [1] by the Lawrence Livermore National Labs, allowing physicists and computer vision professionals the opportunity to revalidate fundamental nuclear explosion models. One goal of the community is to reduce the error in estimating the yield. The error can be reduced significantly by transitioning from a radius-based model to a volume-based model. Unfortunately, no 3D representations exist for atmospheric nuclear detonations, and their testing is now prohibited by treaty. The aim of this paper is to explore a method that could be applied to the past nuclear detonations recorded in the early nuclear age, then 3D representations could be built and yield models improved.

3D reconstruction is a used in computer vision to build 3D models from 2D representations. 3D reconstructions have been built from large collections of photographs [2]. 3D models have also been built from continuous video of a structure using a technique know as structure from motion [3]. Both of these models use scale-invariant feature transform (SIFT) [4] as their method for feature detection. SIFT has become the defacto-standard for image features however, to work well it needs continuous angular coverage. Large gaps in coverage make it difficult and sometimes impossible for features to correlate from different viewpoints. Such is the case with the nuclear detonation (NUDET) videos. Videos are separated by 70-80 degrees in some collections extending beyond the capabilities of SIFT feature matching. As set of features needs to be defined for the NUDETs that can be repeated and used for all stages of the explosion, and correlate between wide angular collections of 70-85 degrees.

One possible feature that could be used in correlating images together is known in the community as “hotspots.” Shown in Fig. 1 they are spots in the explosion that are believed to be debris of the bomb casing. These hotspots exist all over the face of the detonation and their size, orientation, and collective pattern could potentially be used as a feature used correlate and synthesize multiple frames from the same or different viewpoints. To detect the hotspots, 300 features will be defined and data extracted on a collection of pixels from the image. Four kernel sizes will be tested, along with image filtering, dimensionality reduction algorithms and a classifier to learn which features are most effective in detecting hotspots. The top features will then be used on other images to determine how effective the features and classifier are in detecting hotspots.

[*] The views expressed in this paper are those of the authors, and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government. This document has been approved for public release.

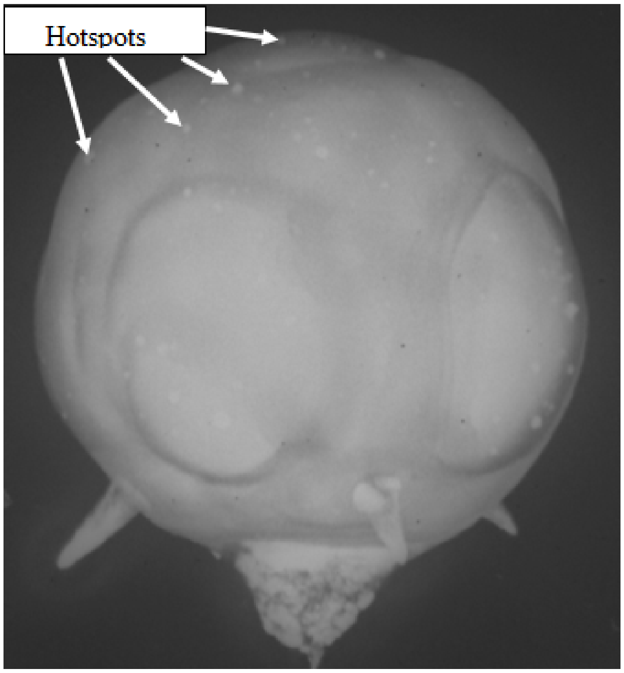


Fig. 1: Tesla explosion [1] with some "hotspots" identified.

The remainder of this paper is organized as follows. Sec. 2 discusses the background of the NUDET films and the dimensionality reduction techniques. Then, Sec. 3 discusses how the experiment is set up. Sec. 4 presents the results and Sec. 5 is the conclusion and future work.

2. BACKGROUND

2.1. Dimensionality Reduction

Dimensionality reduction is the process of reducing a dataset to its most important features to detect a specific item or object. Generically, when combined with a classifier it can be used to classify objects in one of several types, however in its simplest form it can be used to detect the presence or absence of an object. There are two kinds of dimensionality reduction, feature selection and feature extraction. Feature selection is interested in reducing the set of features from an already given set. Feature extraction is interested in finding a new set of dimensions through combinations of the old feature set [5]. Two feature selection techniques, forward subset selection and ReliefF, and one feature extraction technique, principal component analysis (PCA), will be used in reducing dimensionality in detecting the hotspot features.

2.1.1. Forward Subset Selection

Subset selection is focused on finding the best subsets of a set of features. Forward subset selection does this by incrementally adding features and measuring the amount of error that is decreased by adding that feature. Each feature is added, one at a time, and the one that decreases the error the most is kept for future iterations [5]. The drawbacks of using forward subset selection is that, because of its Greedy nature, it is not guaranteed to find the optimal subset, only that each feature that's added is the next best single feature. It also has computation time of $O(d^2)$ for d dimensions.

2.1.2. ReliefF

ReliefF [6] is a special case of Relief algorithm that is a supervised learning feature selection technique that ranks features for most relevant to least relevant. It does this by randomly picking points and measuring the nearest in-class and out-of-class distances. It then calculates a weighting factor for the features to determine which have the greatest weight. ReliefF improves on the Relief algorithm by adding a k-nearest neighbors to the selection reducing the effect of noise [6].

2.1.3. Principal Components Analysis (PCA)

Principal components analysis is an unsupervised method of determining which combinations of features accounts for the greatest variance of data. Different than the last two feature selection techniques, this feature extraction technique seeks to project all the features on a new set of features that are designed to capture the greatest amount of variance in the original set [5].

2.2. Histogram Equalization

Histogram equalization [7] is a process of transforming an image by spreading out the histogram of the pixel intensities of the image in order to maximize how diverse the pixel intensities are. It can be used to prepare an image prior to feature detection to ensure several images meet similar contrast levels. Referring to Fig. 2, the histogram of the original pentagon image [8] is centered in the middle of the 256 gray-scale spectrum. By running a histogram equalization algorithm on the image, the spectrum is spread wider. The result in the image is greater contrast in the coloring. A greater contrast in an image is often beneficial in detecting features because it widens the distance from the pixel values which aids in detecting the distances between those pixel values. It also standardizes that images that are input into the detector reducing the error potential caused by overly dark or light images.

3. EXPERIMENTAL DESIGN

The design of the experiment is set up to determine whether a supervised learning tool like dimensionality reduction is effective in detecting hotspots in the NUDETs. Furthermore, if it is effective, the experiment determines what combinations of parameters and features produce the best results.

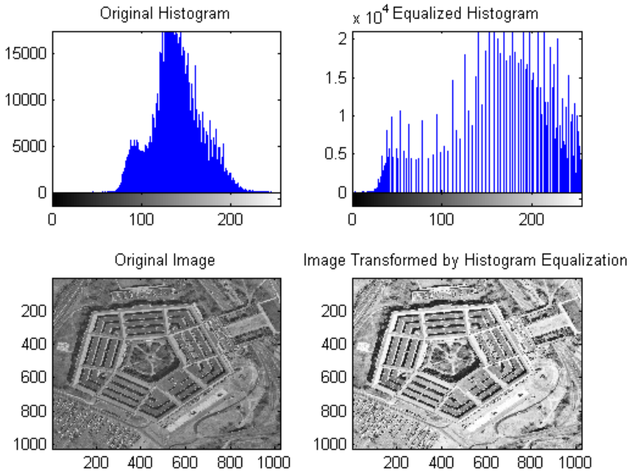


Fig. 2: Histogram equalization applied to a sample image [8].

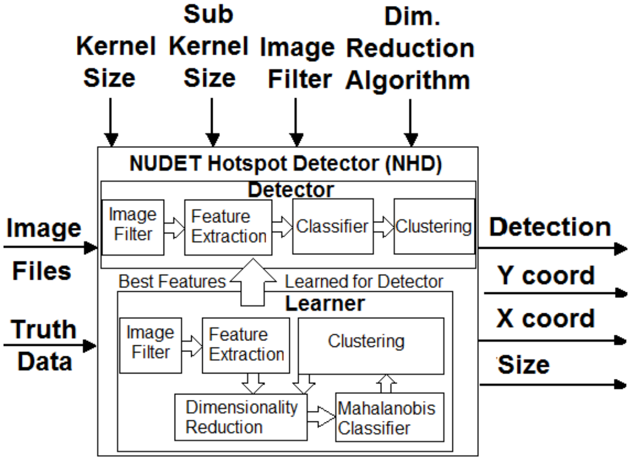


Fig. 3: Block Diagram for hotspot Detector

3.1. Nuclear hotspot Detector

Fig. 3 shows a block diagram for the system that is designed to detect hotspots in a NUDET image aptly named the NUDET Hotspot Detector (NHD). Each of the parameters discussed in this algorithm synopsis is described in detail in Sec. 3.3. The NHD is broken into two primary parts: the learner, and the detector. The learner's function is to use the truth information and dimensionality reduction algorithms to determine which features are best for the detector to use. The learner takes an image in and applies a filter to the image. Iterating through each pixel in the image, the feature extractor extracts 300 features (see Sec. 3.5) from the surrounding pixels based on the kernel size. It then applies a dimensionality reduction algorithm to the features to determine what the most relevant features are. Using the most relevant features, a Mahalanobis-based Classifier determines if the pixel in question is a hotspot or not. Lastly, the detected pixels are clustered into neighboring detected pixels to create the list of hotspots with corresponding locations and size. When the learner is complete, the best features are then used to be the basis of the detector to detect the hotspots. The detector then runs similar to the learner without the dimensionality reduction. It uses the dimensionality reductions recommended features to determine which pixels have hotspots in them.

3.2. Inputs

Along the left side of the block diagram (Fig. 3) are the inputs, which are the subject image and a human interpretation of where hotspots exist in the image (i.e truth data). The truth data is used initially in the dimensionality reduction to "learn" which features are the best features to use in detection. Later, in validation and testing, it is used to determine whether a detection is a true positive or false positive.

Table 1: Kernel sizes tested

Kernel Size	Sub Kernel Size	Num. Sub Kernels
7x7	1x1	49
15x15	3x3	25
21x21	3x3	49
25x25	5x5	25

3.3. Parameters

Along the top of the block diagram (Fig. 3) are the parameters that will be varied in the experiment. The kernel is the size of sub image that will be used. In exploration of the data, it appeared that hotspot size ranged from 5x5 pixels to 14x16 pixels in size. To accommodate the varied size kernels of 7x7, 15x15, 21x21, and 25x25 pixels were defined. The sub-kernel size defined the size of subsections within the kernel. Sub-kernel sizes included 1x1 pixels, 3x3 pixel, and 5x5 pixels related to the kernel size. These are shown in Table 1. The combination of the kernel and the sub-kernel sizes creates a patchwork of sub-kernels that are used to define features (see Sec. 3.5). The image filter parameter is whether preprocessing is used to modify the image prior to feature extraction. Images were either preprocessed with a histogram equalization algorithm or were not filtered. The last parameter is which dimensionality reduction algorithm was used to select the best features for classification. Four dimensionality reduction algorithms were tested: principal component analysis (PCA), forward subset selection, ReliefF, and FCBF(Fast Correlation-Based Filter).

3.4. Outputs

The outputs of the NSD include whether a detection occurred, at what X and Y pixel, and the total clustered size of the detection (in pixels).

Table 2: Number of features per method

Method Grouping	Number of Features
Avg, Min, Max Pixel Values	43
Entropy	12
Prewitt Edge Detection	38
Sobel Edge Detection	38
Zero-cross Edge Detection	38
Canny Edge Detection	38
Gradient X direction	38
Gradient Y direction	38
2D Fourier Transform	17

3.5. Features

Features were defined based on the pixel intensity within a kernel-sized patch of pixels. The goal was to use a variety of methods that would be helpful in detecting hotspots as well as avoiding false detections. Features were grouped by primary method of data source to include edge detection algorithms, the 2D Fourier Transform, entropy, gradient and sub-kernel pixel intensities. A breakout of the number of features summarized by method is shown in Table 2. Since the algorithm needed to work in several stages of the detonation, it was believed that sub-kernel patches would need to be compared to one another. As a result, a series of differencing schemes were defined to compare sub-kernel patches within a kernel. An example of schemes that were used is shown in Fig. 4. Comparing Table 2 and Fig. 4 it can be noted that the same 38 schemes are used for different edge detections and gradient values. Those same 38 schemes are also a subset of the 43 that are used in the average pixel values.

3.6. Workload

Images from the Tesla detonation [1] were used for the workload. Images were chosen from 3 different points of view at 3 different stages of the explosion (early, middle, and late). The first point of view was used to learn, the second to validate the learning, and the last to test and report results.

4. RESULTS

Early in testing it was apparent that the 7x7 kernel size was significantly under-performing with regard to maximizing true detections and minimizing false detections when compared the other three kernels, as a result it was removed from testing early on. This makes sense because it is difficult to detect a 14x16 pixel sized hotspot when only looking at a 7x7 pixel subset of it. Also, FCBF results were discarded. While the speed of FCBF was fast (several seconds), the result was only one or two features which was deemed to be too few to apply to validation and testing data sets.

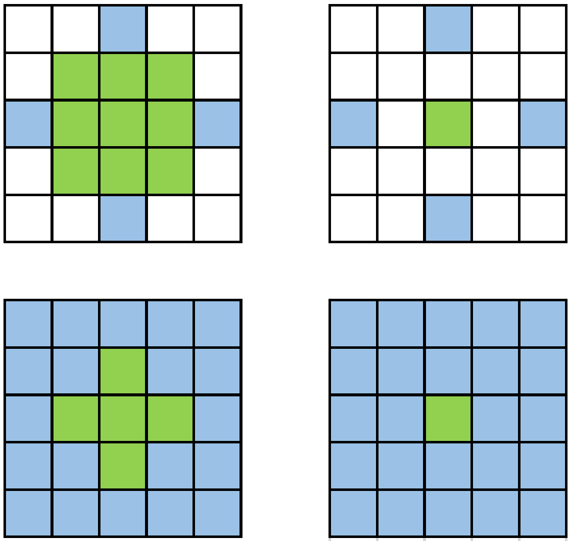
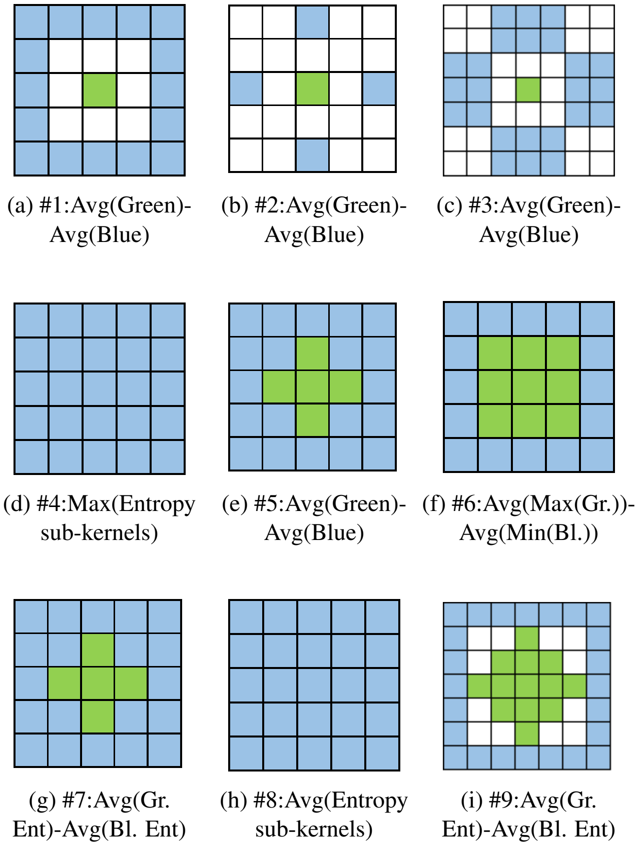


Fig. 4: Four examples of sub-kernel schemes. The 5x5 grid represents 25 sub-kernels for a 25x25 or 15x15 kernel. In the 25x25 case, each square represents a 5x5 pixel area, while in the 15x15 case each square is a 3x3 pixel area. An averaging feature value example would be, averaging the green pixels and subtracted them from an average of the blue pixels. A min/max feature example would take an average of the max of the green sub-kernels minus an average of the min of the blue pixels.

4.1. Best Features

Fig. 5 shows the top few features according to the sequential forward search algorithm. These results are a rough conglomerate of a combination of the six kernel-filter combination results (15x15, 21x21, 25x25 kernels crossed with histogram equalization or not). The top four showed commonality amongst the results identifying the feature in the top two features in at least two different kernels. These results generally make sense for the type of object that we're looking to detect. We would expect a differencing between the center and edges to rank high and they did.

The results of the ReliefF feature selection were very different than the forward subset selection. It took until the 5th feature in sequential forward subset selection with any kernel to match a feature in any kernel in ReliefF (4th feature). Fig. 6 shows the top features that ReliefF found. Again, different that the forward subset selection, there was more consensus amongst the different kernels as to which features were best according to ReliefF. The top 8 features were repeated 6 to 2 times within the top 4 results of a kernel-filter result set. The top feature, the center pixel of the 2D Fourier transform was repeated in the top 3 features of all kernels. Interpreting the results, the top 5 features make sense because they're symmetric both vertically and horizontally, the next several features do not seem to make sense as those particular features



(a) #1:Avg(Green)-Avg(Blue)

(b) #2:Avg(Green)-Avg(Blue)

(c) #3:Avg(Green)-Avg(Blue)

(d) #4:Max(Entropy sub-kernels)

(e) #5:Avg(Green)-Avg(Blue)

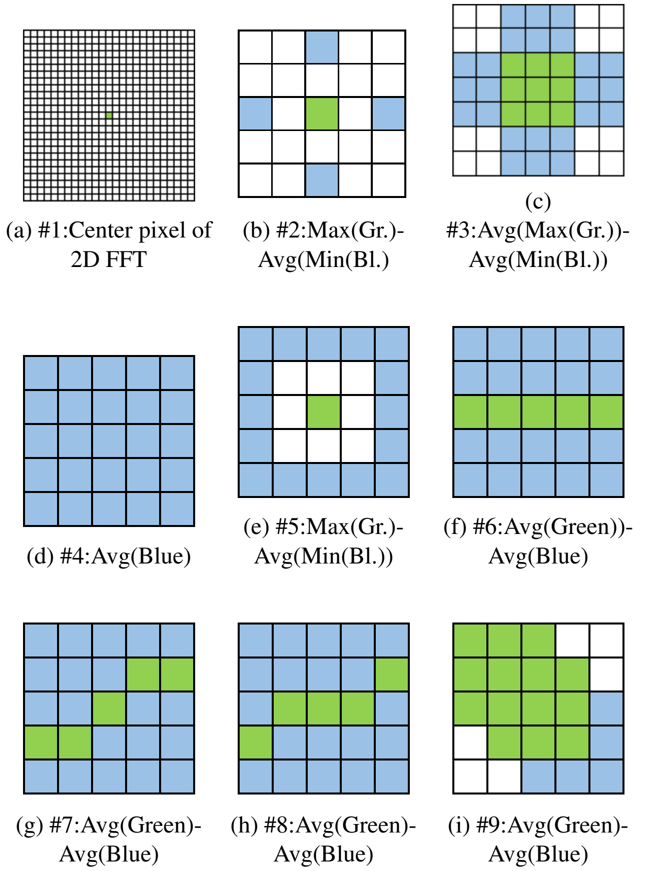
(f) #6:Avg(Max(Gr.))-Avg(Min(Bl.))

(g) #7:Avg(Gr. Ent)-Avg(Bl. Ent)

(h) #8:Avg(Entropy sub-kernels)

(i) #9:Avg(Gr. Ent)-Avg(Bl. Ent)

Fig. 5: Overall best features according to sequential forward subset



(a) #1:Center pixel of 2D FFT

(b) #2:Max(Gr.)-Avg(Min(Bl.)

(c) #3:Avg(Max(Gr.))-Avg(Min(Bl.))

(d) #4:Avg(Blue)

(e) #5:Max(Gr.)-Avg(Min(Bl.))

(f) #6:Avg(Green))-Avg(Blue)

(g) #7:Avg(Green)-Avg(Blue)

(h) #8:Avg(Green)-Avg(Blue)

(i) #9:Avg(Green)-Avg(Blue)

Fig. 6: Overall best features according to ReliefF

were designed to detect other kinds of objects in the image.

4.2. Best Algorithms and Kernels

Fig. 7 shows a summary the hit rate, while Fig. 8 shows a summary of precision and Fig. 9 shows false alarms for all algorithms and kernels tested. Hit rate showed a good spread of capabilities with sequential forward subset (SFS) generally being on top. False alarm rates were very low because of a high number of true negative detections influencing the calculation. ReliefF showed higher precision than most other algorithms. These numbers alone do not tell the full story however.

One obstacle that needed to be overcome to evaluate the effectiveness of the learner and detector was reconciling the truth data. The truth data was designed with a human picking out the center pixel of a hotspot, however, the detector detected the entire hotspot. The results were good and as expected, but the detections needed to be clustered in order to accurately calculate detection probabilities. The effect of the impact of clustering is demonstrated in Fig. 10. In addition, there are some clusters that are rather large (500-1000 pixel area) that need to be rejected, as well as small hotspots (less than 5 pixels) that also need to be rejected. This combination of clustering and rejecting large clusters appears to work best with SFS.

ReliefF results differ greatly from SFS. Fig. 11 shows the unclustered results of ReliefF on a 25x25 kernel with the top 3 features. Fig. 11 shows that the algorithm did not successfully detect individual hotspots, instead it detected the outer region of the explosion. This is also the reason why ReliefF dominates the precision results (see Fig. 8)

PCA results show an improvement in both hit rate (Fig. 7) and precision (Fig. 8) as more features are used. This makes sense for PCA as each additional feature that is added is a finer tuning of the previous features. Comparing 12 features with 24 features, while the 24 feature PCA results have higher hit rate and precision than 12 features, it is possible that it might fall victim to overtraining when applied in a larger scale. The results also show that PCA applied to a histogram equalized 21x21 kernel performs well. A summary of what is considered to be the top performing algorithms is shown in Table 3. A summary of the runtime performance is shown in Table 4.

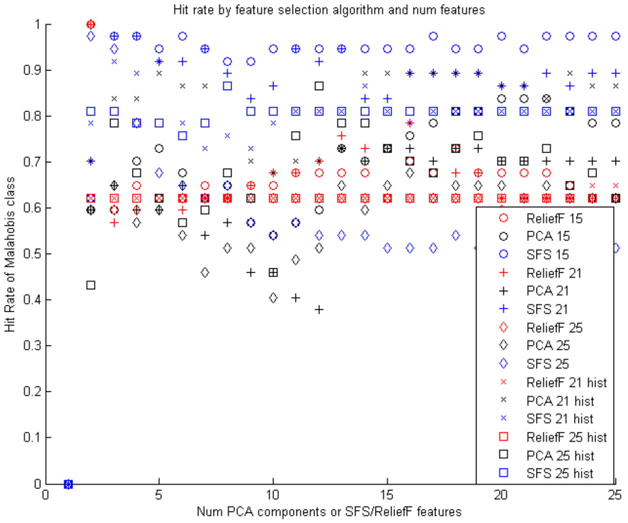


Fig. 7: A plot of measured hit rate results in detecting hotspots with all algorithms attempted, kernel size, and increasing number of features. In general, the SFS algorithms performed best, with PCA algorithms improving as components were added.

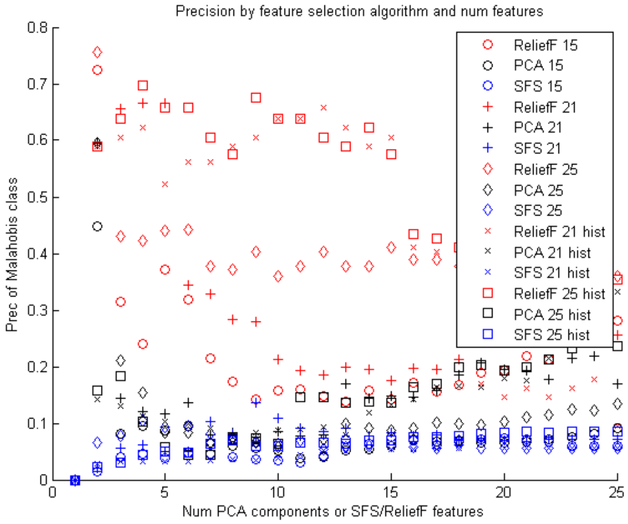


Fig. 8: A plot of measured precision results in detecting hotspots with all algorithms attempted, kernel size, and increasing number of features.

Table 3: Summary of top performing algorithms

Alg.	Kernel	feat.	Hit Rate	Prec.	F.A. Rate
SFS	15x15	3	97.3%	3.95%	0.59%
SFS	25x25	3	97.3%	5.5%	0.41%
PCA	25x25 hist	12	86.49%	14.68%	0.11%
PCA	21x21 hist	5	89.19%	10.25%	0.21%
PCA	21x21 hist	25	89.19%	29.73%	0.047%

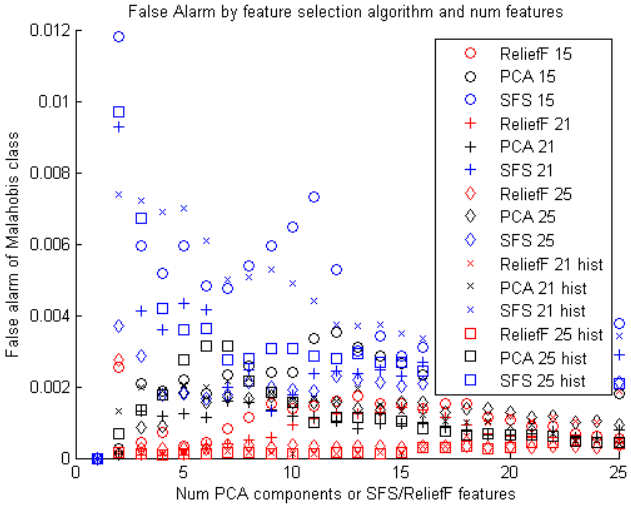
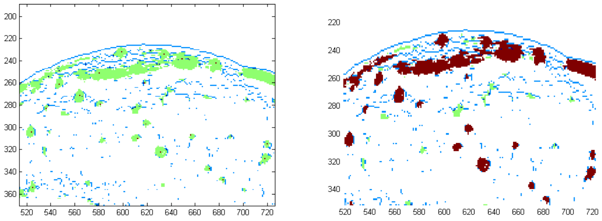


Fig. 9: A plot of false alarm results in detecting hotspots with all algorithms attempted, kernel size, and increasing number of features.



(a) without clustering

(b) with clustering

Fig. 10: A comparison of truth clustering effects on the results of SFS 25x25. Red represents pixels that are true positives. Light green are pixels that are false positives. Without clustering, the left figure has true positives (red dots) surrounded by false positives that is corrected in the right figure.

Table 4: Runtime of algorithms

Algorithm	Avg. Run time per image
Feature Extraction	5.49 hrs
PCA	63 sec
SFS	8.1 hrs
ReliefF	33.2 hrs
Histogram equalization	17 sec

Weighing all combinations of kernels, histogram equalization and dimension reduction algorithms there is no clear top performer, but there is a group of good performers and bad performers. Bad performers include the 7x7 kernel, and the reliefF algorithm for this domain. Good performers include 15x15 and 25x25 kernels for SFS, and histogram equalization paired with PCA for 21x21 and 25x25 kernels. Referring to

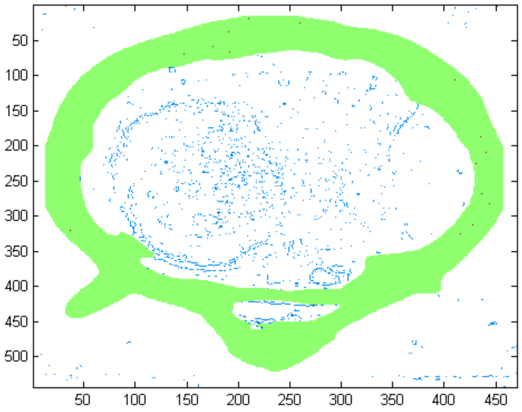
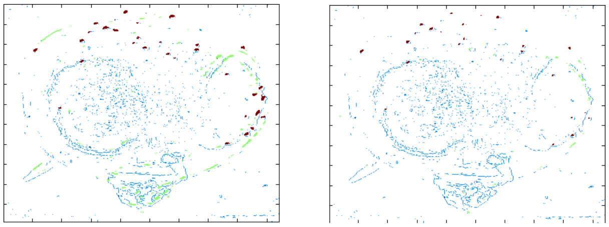


Fig. 11: Results of detections on for ReliefF, 25x25 kernel, top 3 features plotted on an image. Red represents pixels that are true positives (visible inside the light green band). Light green are pixels that are false positives. Clustering turns all the false positives into one large true positive around the exterior of the explosion, and misses true positives in the center of the explosion.



(a) PCA 12 features

(b) PCA 24 features

Fig. 12: A comparison of truth clustering effects on the results of PCA 25x25 kernel with histogram equalization. Red represents pixels that are true positives. Light green are pixels that are false positives.

Table 4, PCA was clearly the fastest algorithm of the dimension reduction algorithms, with SFS next and finally ReliefF.

5. CONCLUSIONS AND FUTURE WORK

In this paper, several combinations of kernels, histogram equalization, and dimension reduction techniques were tested to determine their performance in detecting hotspots in nuclear images. Overall, PCA was the fastest while also maintaining a 86-89% hit rate, 10-30% precision rates, and 0.047-0.11% false alarm rates by applying a histogram equalization filter. SFS, while not performing as quickly as PCA also had 97.3% hit rate with a 0.5% false alarm rate.

For future work, it will be investigated whether these results extend to other features of interest in the explosion like subspheres and cone-shaped protrusions. The results of this experiment will be applied to more datasets to determine if the results are good enough to be a baseline feature detection to aid in 3D reconstruction. It will also be investigated how much the runtime for feature extraction improves by reducing the number of features to just the key features. Lastly, because the histogram equalization shows some benefits with PCA, other filters will be considered and tested like the Wiener filter and median filter.

6. REFERENCES

[1] Spriggs Gregory D., “Film Scanning Project,” Lawrence Livermore National Laboratory, Aug 2011.

[2] Sameer Agarwal, Noah Snavely, Ian Simon, Steven M Seitz, and Richard Szeliski, “Building rome in a day,” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 72–79.

[3] Jan J Koenderink, Andrea J Van Doorn, et al., “Affine structure from motion,” *JOSA A*, vol. 8, no. 2, pp. 377–385, 1991.

[4] David G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Jan. 2004.

[5] Ethem Alpaydin, *Introduction to Machine Learning, 2nd edition*, The MIT Press, Cambridge, MA, 2010.

[6] Yijun Sun and Jian Li, “Iterative Relief for Feature Weighting,” *Proceedings of the 23rd International Conferene on Machine Learning*, pp. 913–920, 2006.

[7] Rafael C. Gonzales and Richard E. Woods, *Digital Image Processing, 3rd edition*, Pearson Prentice Hall, Panchsheele Park, New Delhi, 2009.

[8] “USC-SIPI Image Database,” [Online.] Available: http://sipi.usc.edu/database/database.php .