Automatic Detection of Exploding Aggregates in a Colloidal Suspension

Benjamin D. McPheron, Vincent H. Crespi, Robert T. Collins, and Yanxi Liu
The Pennsylvania State University
University Park, PA 16802, USA
{mcpheron,vhc2}@psu.edu, {rcollins,yanxi}@cse.psu.edu

Technical Report CSE 14-002, Jan. 29, 2014

Abstract

Aggregates in a colloidal suspension of silver chloride in an aqueous hydrogen peroxide solution present behavior of formation and explosion. In an effort to understand this behavior, a novel method for automatic explosion detection on a single frame is developed in this paper. We formulate this problem as a blob-based classification problem. An explosion occurs when large blobs break into smaller blobs and expand outward from a central point. Blob detection is used to locate regions of interest, and binary classification is performed on local features extracted from those regions. To validate the proposed method, our experimental results on nearly 16,000 blobs extracted from real images show high performance rates (classification, sensitivity, and specificity) during cross-validation as well as on image frames from a completely new aggregation video.

1. Introduction

A colloidal suspension of silver chloride in water displays a cyclical process of aggregation and expansion/explosion. In terms of the underlying nonlinear heterogeneous chemical reaction dynamics, this behavior is not well understood. Questions remain about the motion of aggregates, as well as the timing and cause of aggregate explosions. A useful piece of information that can shed light on this process is correlation between exploding blobs based on relative time and spatial separation. Aggregates must be identified within the video frame, along with their center, radius and relevant features, and classified as exploded or intact. Detection of explosions is inefficient when performed manually, as it is difficult to accurately determine the frame at which the blob begins to expand, as well as to quantify the precise blob center and radius.

In an effort to streamline the detection of explosions for high throughput performance, an automatic method has been developed. We propose a novel formalization of this problem as a blob-based binary classification problem. The goal of this work is to determine an good classifier which produces a high rate of correct explosion detection and a low false negative rate. In addition, the effect of training class distribution on the sensitivity and specificity of the classifier is explored.

The data used for this work is a video of a colloidal suspension of silver chloride in an aqueous hydrogen peroxide solution consisting of 752 frames captured at 30 frames per second. The first row Figure 1 shows an example of several consecutive video frames. The second row of Figure 1 shows these same frames with aggregates identified. Blue circles represent an exploded aggregate and red circles represent an intact aggregate. Blob detection is used to locate interest regions containing aggregates. Once interest regions are located, local features are extracted. Previous work using blobs to determine interest regions has shown positive results for brain tumor detection [1, 2], a similar concept is applied in this work.

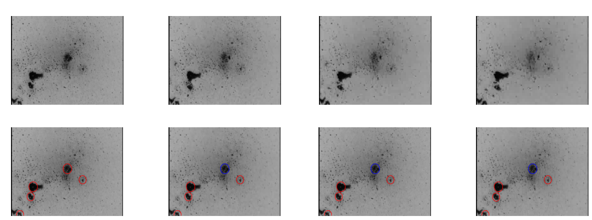

Figure 1: Input output pairs of video frames. The top row shows the original frames, and the bottom row shows frames with labeled blobs.

Blobs have been used in other computer vision applications, including tracking. Statistical models for multi-blob tracking have been developed [3]. Collins [4] showed that blobs could be tracked through scale space using the mean shift algorithm. Blobs have also been used to track multiple humans in a crowded environment [5]. The relevance to blob tracking motivates the development of new ideas for working with blobs.

This work is important to the physics and materials community, as it will enable a study of correlations between an exploding aggregate and its neighbors. In addition, evaluation of extracted features can lead to insights into what physically constitutes an explosion. What we learn from here may also provide methods for better crowd tracking algorithms. This work provides a method that can be extended to identifying grouping and group dispersion in crowds.

The remaining sections of this paper present a method for automatic detection of exploding aggregates. Section 2 presents a formulation of the classification problem. Section 3 describes the classification techniques and motivates studying the effect of class distribution on classifier performance. Results are presented in Section 4, and in Section 5 their implications are discussed, along with future work to improve performance.

2. Problem Formulation

In order to classify blobs as exploded or intact, the interest points must be first located, and then features must be extracted from those interest points. Section 2.1 describes the blob detection process, and Section 2.2 describes and characterizes the features used for classification.

2.1. Blob Formation

Interest regions are areas containing aggregates, which are detected by employing multi-scale Laplacian of Gaussian (LoG) filters to locate blobs of different scales [4, 6]. The blob extraction algorithm is set to capture dark blobs on a light background. Figure 2 shows the blobs found by this algorithm [6].

This algorithm detects a significant number of overlapping blobs. Features should be only be extracted from one blob in a given neighborhood, in order to have a consistent feature set for each local region. To overcome the problem of blob overlaps, the following method of non-maximum suppression is applied to the inverse image of light blobs on a dark background. One blob B_s is selected from overlapping regions by evaluating

$$B_s = \text{argmax}_{C_i}(B_{int}(C_i)) \tag{1}$$

where B_int is the peak intensity of the blob, and C_i represent the blob locations $s.t.$

$$\|C_i - C_k\| < R_i/\alpha \tag{2}$$

for $i \neq j$. Here, R_i is blob radius for the blob located at C_i, and α is a parameter controlling neighborhood size. The challenge posed to performance is tuning the α parameter. If the neighborhood size is selected so that $\|C_i - C_k\| < 2R_i$, then all overlapping blobs that have radii less than or equal to R_i are removed, leaving only the highest intensity blob. Thus, α is set to be 0.5. Figure 3 shows the remaining aggregate interest regions after overlapping blobs are removed and the strongest intensity scale is chosen.

2.2. Feature Extraction

Upon detection of blob interest points blob-based features are assigned. Feature vectors are composed of the following:

- Blob radius
- Blob σ LoG parameter
- Blob peak magnitude
- 8 bin histogram of blob pixel intensities
- Mean of blob pixel intensities
- Variance of blob pixel intensities
- 2 bin histogram of left half blob pixel intensities
- 2 bin histogram of right half blob pixel intensities
- 2 bin histogram of top half blob pixel intensities
- 2 bin histogram of bottom half blob pixel intensities

The last eight feature values are represent simple asymmetry information to determine if the explosion occurs in a specific direction.

Of the features, it is found that the highest bin of pixel intensities, containing values from 224-255, is always zero, so this feature is removed. The remaining features are used to form feature vectors of length 20. With 752 frames of video, and an average of 21 detected blobs per frame, the data set is composed of 16,000 blob-based feature vectors. Each of these blobs is hand labeled as either exploded (Class 1) or intact (Class 0). Figure 4 shows eight example frames with hand labeled data. From these frames it is apparent that the number of exploded blobs is much less than the number of blobs that are intact. An evaluation of the data set reveals that the probability of a blob being exploded is $P(1) = 0.1323$ and the probability that the blob is intact is $P(0) = 0.8677$ among the extracted blobs.

Figure 5 shows close up examples of each class. Intact blobs display a higher concentration of darker pixels and smaller blob radii than exploded blobs.

Fisher's criterion [7] is used to rank the top discriminative features for classification. Fisher's criterion is used to evaluate the signal-to-interference ratio between two classes. This criterion is defined by

$$J(w) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2} \tag{3}$$

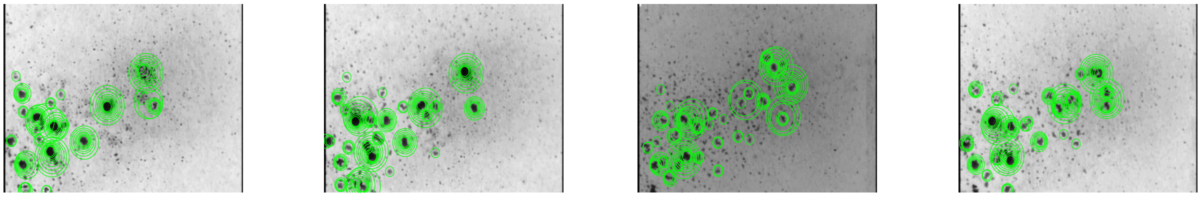

Figure 2: Video frames with blobs detected.

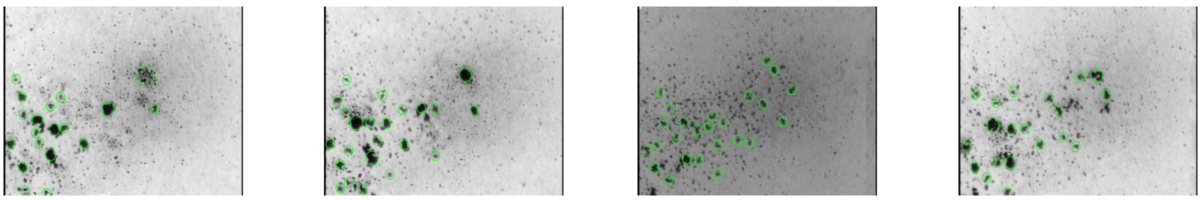

Figure 3: Video frames with overlapping blobs removed.

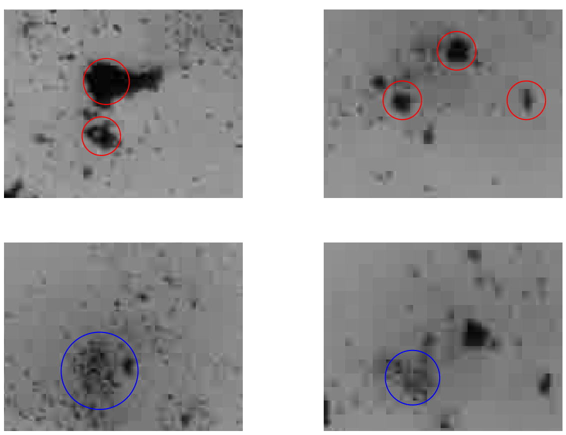

Figure 5: Close up examples of intact and exploded classes.

where m_1 and m_2 are the means of the two classes, and s_1^2 and s_2^2 are the their variances. The higher the value found by this function for a given feature, the higher the separation between the classes with respect to the feature. Using this criterion, the top three single features most important for classification are found to be the blob radius, the mean of pixel intensities, and the pixel intensity bin with values 32-63. Table 1 shows the Fisher criterion results normalized by the highest value.

The data distributions for the two classes on these top three features can be seen in Figure 6. These distributions show that data is not linearly separable in one dimension for any of the top-three features. Thus, a higher dimensional feature space is needed. Data can visualized in the

Feature	Rank	Normalized Fisher
Blob Radius	1	1.0
Mean Intensity	2	0.58
Intensity Bin 32-63	3	0.36

Table 1: Top three features ranked using Fisher's criterion

top-three feature space to get a better understanding of class separation. Figure 7 shows data points plotted in the top three feature space. It is clear that the intact data points are clustered together in this space, with a few outliers. The outliers arise from data points that are intact but have very large radii. This cluster relates directly to a period of the video in which a large aggregate forms before exploding. It is apparent that most, but not all, exploded blobs have larger radii than intact blobs.

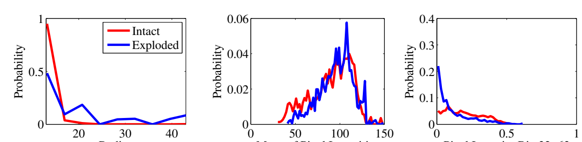

Figure 6: Distributions for top three features.

3. Classification

Three explosion detection classifiers are trained on training sets with two different class distributions. Section 3.1 provides a brief explanation of each classification method. Section 3.2 describes the motivation of balancing class distributions for classification.

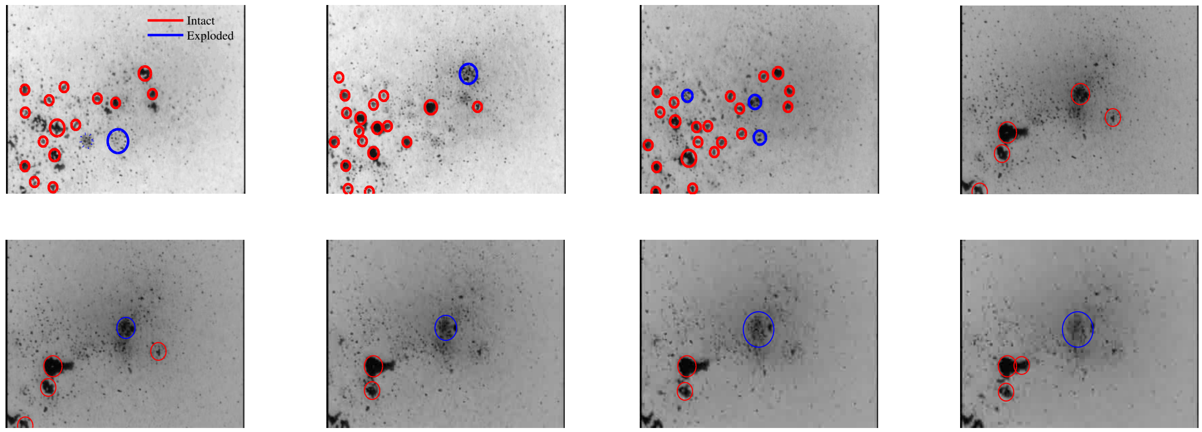

Figure 4: Video frame with blobs labeled as Class 0 (intact) or Class 1 (exploded).

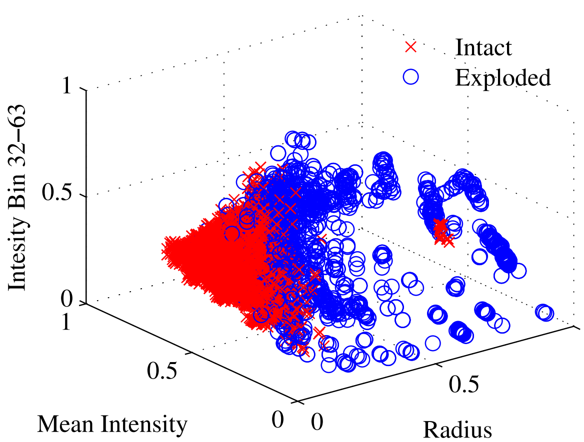

Figure 7: Visualization of all blobs in the feature space spanned by the top three features ranked by Fisher's Criterion.

3.1. Classification Methods

The simplest method applied for classification is the Linear Least Squares (LLS) classifier. The goal of this method is to generate an optimal weight vector $\mathbf{w}^*$ to provide the best classification boundary using ordinary least square fit for N sample points. This method is expected well only when the data is linearly separable [8].

A second method considered for classification is Maximum Likelihood Estimation (MLE). In MLE, rather than learning a decision boundary, a predictive distribution is formed from the training data. The optimal parameter $\hat{\mu}$ is found as the value maximizing the joint likelihood of N training samples [8].

The final method used for classification is Random Forests [9, 10]. A random forest is a collection of classifiers that have a tree structure where each tree votes for a class on the input x. The class that has the most votes at the end of this procedure is chosen as the 'winner', and selected as the class for the data. This collection of tree classifiers can be described as$\{h(x, \Theta_k), k = 1, 2, 3, ...\}$ where $\{\Theta_k\}$ are assumed to be independent, identically distributed random vectors [11]. Breiman [11] showed that random forests always converge, and that the accuracy of the classifier can be improved by increasing the number of decision trees.

3.2. Class Distribution

Correctly detecting an explosion and avoiding false negatives are primary goals of this work. Weiss and Provost [12] studied the effect of class distribution on tree induction and found that in some cases the naturally occurring class distribution performs best, and in other cases the balanced class distribution is optimal. This motivates our study of the effect of class distribution on correct detection of explosions and false negative rate. In an attempt to study these goals, training and testing data sets are generated according to two class distributions:

1. Naturally occurring class distribution $P(1) = 0.1323$ and $P(0) = 0.8677$
2. Balanced class distribution $P(1) = 0.5$ and $P(0) = 0.5$

4. Results

4.1. Original Data Set

Several metrics are chosen to quantitatively validate the performance of the classifiers used in this work. The most obvious metric is classification rate. The rate accuracy is defined as the ratio of correctly classified data points to the total number of data points. However, since the naturally occurring class distribution shows only a 13.23% chance of a blob being exploded, the classification rates must be higher than 86.77% to be non-trivial.

Other performance metrics afford more specific evaluations of classifier performance. The performance metrics which are most meaningful to the goals of this work are sensitivity and specificity. Sensitivity is defined as

$$Sensitivity = \frac{\text{Correctly classified as exploded}}{\text{Actual number exploded}} \quad (4)$$

and represents the correct detection rate for exploded blobs. Specificity is defined as

$$Specificity = \frac{\text{Correctly classified as intact}}{\text{Actual number intact}} \quad (5)$$

and represents the complement of the false negative rate. As the goal of this work is to provide a high correct detection rate and a low false negative rate, both the sensitivity and specificity should be high for the optimal classifier.

Cross validation is performed 100 times on randomly split training-testing sets for statistically significant results. For the naturally occurring class distribution, data is split randomly so that 95% of the data is used for training, corresponding to 15800 data points, and 5% of the data is used for testing. For the balanced distribution, significantly less data is used for training, only 4000 data points. The categorical distribution is used for MLE, and 500 decision trees are used for the random forest algorithm to provide a balance between accuracy and computational time.

Table 2 shows the classification accuracy results for the naturally occurring class distribution and the balanced distribution when used by the three classification methods. The LLS and MLE methods do not perform well, since the classification rate is below 87%. The random forest method performs well on both class distributions.

Class Rate	Linear Least Squares	Maximum Likelihood	Random Forest
Training (Natural)	77.1 $\pm$24.4%	75.5 $\pm$ 0.2%	99.6 $\pm$ 0.1%
Testing (Natural)	77.9 $\pm$21.8%	75.0 $\pm$ 3.2%	97.9 $\pm$ 0.4%
Training (Balanced)	67.4 $\pm$14.8%	74.5 $\pm$ 0.4%	99.8 $\pm$ 0.1%
Testing (Balanced)	71.7 $\pm$24.7%	75 $\pm$ 3.2%	94.1 $\pm$ 1.5%

Table 2: Classification accuracy results, defined as the percentage of correctly classified aggregates.

Table 3 shows the sensitivity and specificity results of the Random Forest classification method. It is important to note that a balanced data distribution increases the correct detection rate for explosions by 6%. However, there are more false negatives for a balanced data set, reflected by a lower specificity.

Random Forest	Sensitivity	Specificity
Training (Natural)	97.8 $\pm$ 0.1%	99.9 $\pm$ 0.1%
Testing (Natural)	88.6 $\pm$ 3.3%	99.3 $\pm$ 0.3%
Training (Balanced)	99.8 $\pm$ 0.1%	99.8 $\pm$ 1.0%
Testing (Balanced)	94.7 $\pm$ 4.3%	94.0 $\pm$ 1.7%

Table 3: Sensitivity and specificity results for the Random Forest classification method.

4.2. Application to New Data

To verify the generalizability of the explosion detection algorithm, we apply the Random Forest classification model learned on the natural distribution and balanced distribution from the original data to features extracted from a new video. The classification rate, sensitivity, and specificity are tabulated in Table 4. These results validate that the learned classifier is generalizable.

Random Forest	Classification Rate	Sensitivity	Specificity
Natural	98.4%	90.4 %	99.4%
Balanced	96.5%	83.1 %	98.9%

Table 4: Performance metrics for new data set

A supplementary video included with this work show the results of classification for a sequence of video frames, even though temporal information is not used.

5. Discussion

The results in the previous section show that balancing the class distribution for the random forest classification method provides the benefit of increased sensitivity. This means that an explosion is more likely to be correctly detected in this formulation. However, this comes at the cost of more false negatives, as the specificity is decreased. Balancing the class distributions and reducing the number of training points by an approximate factor of 4 also has the benefit of decreasing computational time for training. Table 5 displays the reduction in computational time for training that is afforded by balancing the class distribution and reducing the number of data points. With these factors in mind, we feel that the increase in computational efficiency and correct explosion detection rate provided by the balanced approach provide a viable method for collecting meaningful data about explosion correlation and the behavior of aggregates in colloidal suspensions.

Distribution	Time [s]
Natural	17.8
Balanced	3.6

Table 5: Computation training time for two class distributions

Several possibilities exist to explain why some blobs are misclassified. Figure 8 shows two blobs misclassified as exploded when they are actually intact. Figure 9 shows two blobs misclassified as intact, when they are in fact exploded. These figures show that for some blobs, the image shows artifacts or distortion. This distortion can lead to the varied contrast or intensity differences, making the blob appear incorrectly exploded or intact to the classifier. Another possible source of error is blob radius. In general it is observed that exploded blobs have a larger radius than intact blobs. However, there exist very large aggregates with large radii, as observed earlier in the top-three feature space in Figure 7. These large aggregates are outliers, and could be identified as exploded in error. A third possible source of error is blob intensity. If exploded blobs contain a majority of very dark pixels, they may be incorrectly classified as intact.

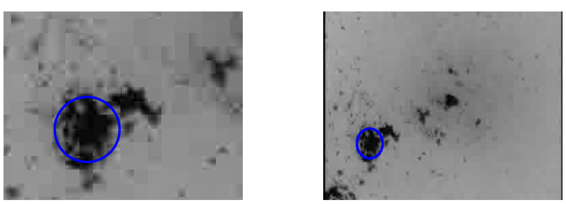

Figure 8: Blobs misclassified as exploded.

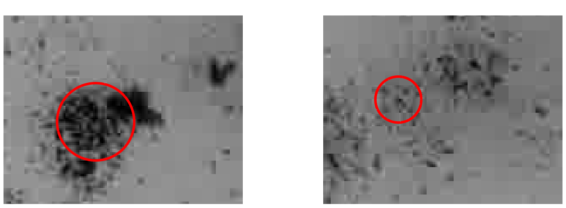

Figure 9: Blobs misclassified as intact.

There may be additional features which can help to improve correct explosion detection and reduce false negative rates. Features under investigation include temporal information about the interest regions and the introduction of a correspondence factor from one frame to the next. Note that these additional temporal features are particularly promising in that an explosion is intrinsically a temporal phenomenon. Also of interest is the use of texture features within blobs. Exploded blobs show a varied pattern composed of many small spots, while intact blobs have a much more uniform texture. Upon the addition of further features, it will be necessary to perform feature selection and evaluate the features which are most important for effective classification. Evaluating these features will provide insight into the physical information indicating an explosion.

That fact that our proposed method tested on the original data set, as well as directly applied to a new video data set gives us confidence in its generality. The resulting performance shows similar classification rate, sensitivity, and specificity to the original data set. This verifies the potential for applying the novel explosion detection method presented in this paper to other videos of similar problem tasks.

References

[1] D. Koshy, C. Yu, D.T. Nguyen, MD, S. Kashyap, R.T. Collins, Y. Liu, Supervised machine learning for brain tumor detection in structural MRI, *Presentation at Radiological Society of North America* (RSNA) 2011. 1

[2] C.P. Yu, G.C.S. Ruppert, D.T.D. Nguyen, A.X. Falcao, Y. Liu, Statistical asymmetry-based brain tumor segmentation from 3D MR images, *Proc. of the 5th International Conference on Bio-inspired Systems and Signal Processing* (2012) 527-533. 1

[3] M. Isard, J. MacCormick, BraMBLe: a Bayesian multiple-blob tracker, *Computer Vision* (ICCV) 2001 Proceedings, 34-41 Vol 2. 1

[4] R.T. Collins, Mean-shift blob tracking through scale space, *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) June 2003, Vol. 2, 234-240. 1, 2

[5] T. Zhao, R. Nevatia, Tracking multiple humans in a crowded environment, *IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) 2004, Vol. 2, 406-413. 2

[6] T. Lindeberg, Feature detection with automatic scale selection, *Int. J. of Computer Vision* (1998) 79-116. 2

[7] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, *Wiley, New York* 1973. ISBN: 0471223611 2

[8] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), *Springer-Verlag New York, Inc.* Secaucus, N.J. USA, 2006. ISBN: 0387310738 4

[9] T.K. Ho, Random decision forest, *Proceedings of the 3rd International Conference on Document Analysis and Recognition* Montreal, QC (1995) 278-282. 4

[10] T.K. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (8) (1998) 832-844. 4

[11] L. Breiman, Random forests, *Machine Learning* 45 (2001) 5-32. 4

[12] G.M. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, *J. of Artificial Intelligence Research* 19 (2003) 315-354. 4