Hierarchical Disaster Image Classification for Situation Report Enhancement

Yimin Yang, Hsin-Yu Ha, Fausto Fleites, Shu-Ching Chen, Steven Luis
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
{yyang010, hha001, fflei001, chens, luiss}@cs.fiu.edu

Abstract

In this paper, a hierarchical disaster image classification (HDIC) framework based on multi-source data fusion (MSDF) and multiple correspondence analysis (MCA) is proposed to aid emergency managers in disaster response situations. The HDIC framework classifies images into different disaster categories and sub-categories using a pre-defined semantic hierarchy. In order to effectively fuse different sources (visual and text) of information, a weighting scheme is presented to assign different weights to each data resource depending on the hierarchical structure. The experimental analysis demonstrates that the proposed approach can effectively classify disaster images at each logical layer. In addition, the paper also presents an iPad application developed for situation report management using the proposed HDIC framework.

1. Introduction

Due to the ease of access and wide reach of Internet, more and more multimedia data such as images and videos, along with corresponding textual descriptions, become available through the web everyday. Such availability of content-rich data is extremely valuable for emergency management (EM) personnel as they can take more accurate decisions in disaster situations by having both textual and visual information of the disaster. Nevertheless, currently, EM personnel mostly utilize disaster situation reports (also referred as situation reports) which provide just a textual description of the disaster. To augment situation reports with related disaster images and thus provide EM personnel with images and videos that present valuable information about the disaster, a hierarchical disaster image classification (HDIC) framework is proposed in this paper. Based on multi-source data fusion (MSDF) and multiple correspondence analysis (MCA) [1], our framework classifies disaster multimedia data into different categories and links these images to related situation reports. In order to obtain images for the disaster domain (i.e., hurricane, oil spill, and earthquake), we collected both images and their corresponding titles and description from Flickr. The HDIC framework utilizes both visual features from images and textual description to demonstrate the performance of combining MCA-based data fusion method with the hierarchical classification approach.

There are two main applications for image classification in the area of disaster analysis: damage detection and damage prediction. Najab [13] used Principal Component Analysis (PCA) to extract the features from remotely-sensed data and classify them into different landcover classes. Gandhe [8] leveraged a framework which includes discrete wavelet transform (DWT) and PCA to help with image mining and weather forecasting, and Hsu [9] applied wavelet transformation, support vector machines, and fuzzy neural networks for image compression, classification and error correction respectively to an intelligent typhoon damage prediction system. In addition, classification of high-resolution disaster images facilitates the process of damage assessment after environmental disasters such as hurricane, tsunami, etc [12, 6, 2, 3, 4]. Unlike the aforementioned works that focus on satellite images [13, 3, 4], images retrieved from multiple remote sensing sensors [12, 6] and aerial photos [9, 2], our framework is able to classify the actual disaster images taken at the disaster location, which have higher complexity and reduce the semantic gap between the images and the disaster categories. In addition, the proposed framework is able to fusion multi-source data in an efficient way achieving higher performance than the individual textual and visual models independently.

The remainder of this paper is organized as follows. Section 2 briefly describes the HDIC framework based on MSDF and MCA. Section 3 discusses the MCA algorithm for multimedia content analysis. Section 4 presents the details of the visual-text model training. Section 5 discusses the hierarchical classification based on MSDF.

Experimental analyses is presented in section 6, and section 7 briefly introduces the ipad application developed based on the HDIC framework. Finally, section 8 concludes the paper.

2 HDIC Framework

Depicted in Figure 1, the HDIC framework is composed of two main processes: multi-source model training and hierarchical classification. During the model training process, visual and text features are extracted respectively and fused based on the weighting scheme presented in section 5. Then the models for different categories and sub-categories (subjects) are trained based on the MCA algorithm, generating thresholds for classification. The feature extraction of testing data depends on that of the training data. For example, the discretization intervals of test visual feature should corresponds to those of the training data. Finally, the trained models are applied to the hierarchical classification of images, where the images are firstly classified into general categories, and then passed to the next layer to be assigned to specific subjects.

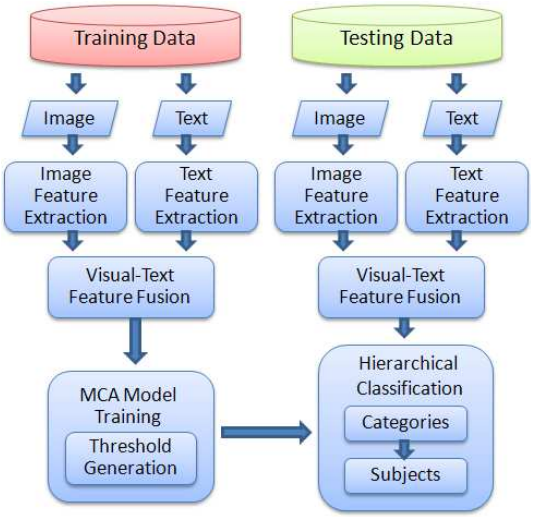


Figure 1: HDIC framework.

3 MCA for Multimedia Content Analysis

MCA is an exploratory data analytic technique designed to analyze multi-way tables for some measure of correspondence between the rows and columns [1]. It is a natural extension of the standard correspondence analysis to more than two variables. The observations used for MCA are a set of nominal variables, each of which is composed of several levels, and each level is coded as a binary value. There is a constraint that one and only one level of the variable gets the value 1. Therefore each observation has the same total, called mass. MCA can also accommodate quantitative variables by recording them as bins, which inspires the idea that it could be applied to numerical data, such as multimedia feature instances. For example, each image feature variable could be discretized into several intervals, and each image can be presented by a series of nominal values.

Motivated by the functionality of MCA as well as its quantitative analysis ability, the utilization of MCA has been explored in our previous works to analyze the data instances described by a set of low-level features to capture the correspondence between items (feature-value pairs) and classes (subjects). The similarity of every item and every class can be presented by the cosine of the angle between each item and class [10, 11]. A smaller angle indicates a higher correlation between the item and class.

4 Visual-Text Model Training Based on MCA

This section reveals the feature extraction processes for both visual and text data as well as the model-training procedure based on the MCA algorithm. An iterative threshold determination algorithm is also presented to find out the most appropriate threshold for classification.

4.1 Visual feature extraction

There are mainly three steps for visual feature extraction: feature extraction, normalization, and discretization. The first two steps are the same for both training images and test images; however, the discretization of the test images' features is based on the discretized intervals resulted from training image instances.

In order to capture the visual contents of images, two types of feature are extracted: low-level color features and mid-level object location features, which are described as follows:

- *Twelve color features*: black, white, red, red-yellow, yellow, yellow-green, green, green-blue, blue, blue-purple, purple, and purple-red;
- *Nine object location features*: Images are divided into 3×3 grid, i.e., nine locations $L1, \cdots, L9$, where $L_i = 1$ if there is an object whose centroid falls inside $L_i = 1, 1 \leq i \leq 9$.

Therefore a total number of 21 features are obtained, where the color features are based on the HSV color space, and the object location features are extracted using the SPCPE algorithm [5]. Since the color features and object location features are considered equally important, an equal weight (i.e., 0.5) is assigned to each type of features in the normalization step. Finally, an information-gain-based discretization method [7] is used for numerical to nominal transformation.

4.2 Text feature extraction

Due to its limitation in descriptive capability, visual features alone could not well represent the content of an image. Therefore text features are introduced to enhance the description. The proposed text feature extraction procedure requires more preprocessing than visual features. First, punctuation characters and stop words are removed, thus obtaining a list of valid words for each image instance. The word frequency is calculated based on all the training instances for each concept (subject). The top N (i.e. 50 in our experiment) words with the highest frequencies are selected as features. A nominal value is assigned to each feature representing the existence or absence of it. Then each image instance could be transformed to a sequence of nominal variables with N dimensions. The feature extraction process of the test data set is almost the same as that of the training data set except for the ”get word frequency” step since the construction of testing feature vector is based on the top N words from training data.

4.3 Visual-Text Model Training

The process of visual-text model training can be summarized into two major steps: MCA score calculation and threshold generation. More specifically, after visual and text feature extraction of the training data sets, the two sets of feature vectors are concatenated together to form a data set of fused instances, which are used for angle generation based on MCA correlation analysis. The angles, denoted as A, are calculated using Equation (1), where I and C are two-dimensional principal components representing items and classes respectively as described in section 3, and j, k are indicators of items and features. Then the generated angles are applied to weight conversion as shown in Equation (2). The weight is a measure of the similarity between each item and class. The sum of all of the weights within one instance is denoted as S (shown in Equation (3)), which is the final evaluation of the relationship between each instance and class. A higher score implies a higher possibility that the instance belongs to the class, which implies the existence of a cut point (threshold) determining the positive or negative attribute of one instance for certain class (subject).

$$A_k^j = arccos(\frac{I_k^j \cdot C)}{\left|I_k^j\right| |C|}), \tag{1}$$

$$weight_k^j = \pm(1 + cos(A_k^j \times \pi/180)), \tag{2}$$

$$S_i = \sum_{k=1}^{K} weight_k^j, i \in \{1, 2, \cdots, N\} \tag{3}$$

How to determine the threshold is a critical issue and plays an extremely important role in the final performance of the whole classification algorithm. Therefore an iterative method is designed to find out the threshold for classification based on the training instances:

```
THRESHOLD-GENERATION:
1 $finalF1 = 0$;
2 $finalThresh = 0$;
3 $sortedScore =$ **sort** $(trainScore)$;
4 $cddThresh =$ **find** $(positive)$;
5 for $i = 1$ **to length** $(cddThresh)$
6     $testLabel$ $(1$ **to** $cddThresh(i)) = classLabel$;
7     $F1$ $calculation$;
8     **if** $finalF1 > F1 \,||\, finalF1 - F1 < \gamma$ **then**
9         $finalF1 = F1$;
10        $finalThresh = sortedScore(cddThresh(i))$.
```

Steps 1 and 2 initialize the variables of $finalF1$ and $finalThresh$, which store the final F1 score and the corresponding threshold. In step 3, the sort function sorts training scores in descending order, and step 4 finds the indexes of positive scores from the sorted array as candidate thresholds. Step 5 through 10 loop through each candidate to find the best threshold giving the optimal performance in terms of the F1 measure. Specifically, steps 6 and 7 calculate the F1 scores based on precision and recall (refer to section 6). In step 8, the latter condition (i.e., $finalF1 - F1$) is designed to include the neglected positive instances; it provides the functionality of balancing between recall and precision measures and improves F1 scores. The term γ is a practical parameter, and it is set to be 0.03 in the experiments. Finally, steps 8 and 9 recall the final F1 score and threshold.

5 Hierarchical Classification

In order to explore the extensive relationship between various subjects and perform the classification in a more efficient way, a hierarchical classification mechanism is proposed. The hierarchical classification scheme breaks down disaster-related categories into a tree structure which serves to organize general to specific categories. The classification scheme addressed in this paper was developed

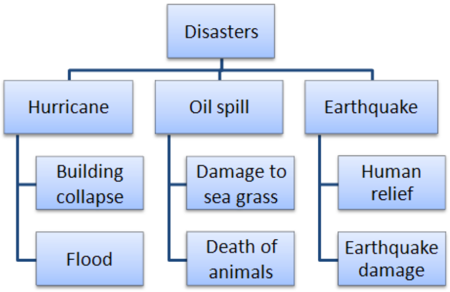


Figure 2: Hierarchical structure.

upon consulting with experts in the disaster management field.

As shown in Figure 2, the top-down category tree classifies images into one of three main categories, i.e., hurricane, oil spill, or earthquake based on text-visual models, and then the chosen category will be further classified into a specific sub-category. Based on the observation that the text data in the second layer has a stronger pattern than that of visual model and vice versa in the third layer, a weighting scheme is proposed to distinguish the significance of visual and text models at different layers and obtain a better fusion result. The fusion score is calculated as follows:

$$score_f = \alpha W_v * score_v + \beta W_t * score_t, \quad (4)$$

$$thresh_f = \alpha W_v * thresh_v + \beta W_t * thresh_t, \quad (5)$$

$$W_v = \frac{F1_v}{F1_v + F1_t}, \; W_t = \frac{F1_t}{F1_v + F1_t}, \quad (6)$$

$$W_v + W_t = 1, \; \alpha + \beta = 2. \quad (7)$$

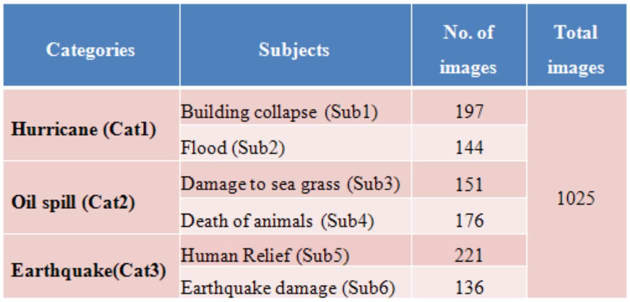
where $score_v$ and $score_t$ represent the scores obtained from visual and text models, while αW_v and βW_t denote the weight factors of visual and text models respectively, and $score_f$ is the final fused score. The thresholds are fused in the same manner. The W_v and W_t are calculated based on the F1 measures of visual and text models at different layers, while the α and β are tuning parameters. In the experimental analysis, the α and β are set to be 0.50 and 1.50 in the second layer; 1.23 and 0.77 in the third layer. Finally, the classification rules are generated as follows:

$$finalLabel = \begin{cases} positive, \; if \; score_f \geq thresh_f, \\ negative, \; if \; score_f < thresh_f. \end{cases} \quad (8)$$

6 Experimental Analysis

In order to demonstrate the effectiveness of the proposed MCA-based multimedia content analysis, a set of experiments have been conducted to evaluate its performance. The test bed is a web-crawled dataset consisting of 1,025 images with texts downloaded form Flickr. The number of images is limited due to the fact that domain-specific disaster images are not abundant. The images contain three categories and cover six subjects as shown in Figure 3. The categories are denoted as Cat1, Cat2, and Cat3, and the subjects are denoted as Sub1 through Sub6.

In the experimental settings, the hierarchical classification scheme shown in Figure 2 is adopted. Multi-source (text and visual) data fusion is performed at both layer 2 and layer 3. To show the advantages of the multi-source model over single–source models, a comparison between the performances of the multi-source text-visual model and the single-source text and visual models are conducted at each layer. The precision (Equation 9), recall (Equation 10), and F1 (Equation 11) are calculated as the measurements of performance under the 3-fold cross validation approach.



Categories	Subjects	No. of images	Total images
Hurricane (Cat1)	Building collapse (Sub1)	197	1025
	Flood (Sub2)	144	
Oil spill (Cat2)	Damage to sea grass (Sub3)	151	
	Death of animals (Sub4)	176	
Earthquake(Cat3)	Human Relief (Sub5)	221	
	Earthquake damage (Sub6)	136	

Figure 3: Composition of categories and subjects.

$$precision = \frac{TP}{TP + FP}, \quad (9)$$

$$recall = \frac{TP}{TP + FN}, \quad (10)$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}, \quad (11)$$

where TP, FP, and FN represent the number of true positive, false positive and false negative instances respectively. Tables 1 through 3 show the performance evaluation results for layer 2. Specifically, tables 1 and 2 give the scores of text and visual models respectively, and table 3 shows the results of the fused model. As shown in the tables, the fused model outperforms the single-source models. The visual-text model approach achieves a 7% improvement over the text model and a 36% over the visual model. Another observation is that the text model outperforms the visual model. This is because the text information at layer 2 shows a stronger pattern than that of visual information. For example, there is a high possibility that the text files describing images of Cat1 contain the key ”hurricane”, while the text files belonging to Cat2 contain the words ”oil” and ”spill”. However, the visual contents of the corresponding images are more abstract and complicated, especially when many categories and subjects are involved. Therefore, a higher weight is assigned to text features at layer 2.

The advantages of text features diminish gradually as the categories are further classified into specific subjects since there is not a strong distinction among those text files in the same category. On the other hand, the visual features demonstrate their superior characteristics for extracting visual patterns when there are fewer subjects involved. Therefore, the weight for visual features increases at layer 3. Tables 4 through 6 contain the subject classification results of layer 3. Specifically, table 4 and table 5 present the scores of text and visual models respectively, and table 6 shows the performance of the combined model. The categorization results of layer 2 enhance the power of visual model at layer 3. The final F1 score of the whole classification framework is 83%, which is 10% and 5% higher than the visual and text models respectively. Although the performance of layer 3 is not as good as layer 2 due to the error propagation problem, the overall experimental results demonstrate the advantages of the data fusion method based on MCA as well as the effectiveness of the hierarchical classification approach.

Categories	Precision	Recall	F1
Cat1	0.98276	0.99415	0.98839
Cat2	0.82698	0.91743	0.86625
Cat3	0.73662	0.97199	0.80158
Average	0.84879	0.96119	0.88541

Table 1: Performance evaluation for text model (Layer-2).

Categories	Precision	Recall	F1
Cat1	0.4485	0.62405	0.51397
Cat2	0.53	0.69419	0.59718
Cat3	0.60715	0.68908	0.6434
Average	0.52855	0.6691	0.58485

Table 2: Performance evaluation for visual model (Layer-2).

Categories	Precision	Recall	F1
Cat1	0.98825	0.98533	0.98678
Cat2	0.9578	0.90214	0.9291
Cat3	0.94653	0.93277	0.93925
Average	0.96419	0.94008	0.95171

Table 3: Performance evaluation for visual-text model (Layer-2).

Subjects	Precision	Recall	F1
Sub1	0.87877	0.86807	0.86899
Sub2	0.85186	0.82548	0.83193
Sub3	0.94494	0.85296	0.89631
Sub4	0.92468	0.93569	0.92994
Sub5	0.64674	0.76712	0.67544
Sub6	0.43501	0.58087	0.49665
Average	0.78033	0.80503	0.78321

Table 4: Performance evaluation for text model (Layer-3).

Subjects	Precision	Recall	F1
Sub1	0.64212	0.84219	0.72403
Sub2	0.65548	0.84511	0.73546
Sub3	0.74588	0.80961	0.76942
Sub4	0.75924	0.74552	0.72477
Sub5	0.7355	0.89078	0.79925
Sub6	0.59502	0.75858	0.66468
Average	0.68887	0.8153	0.73627

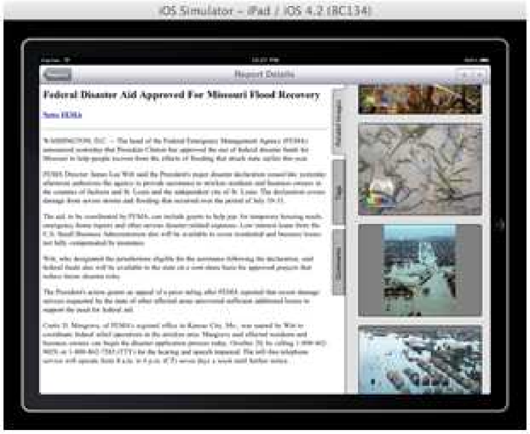
Table 5: Performance evaluation for visual model (Layer-3).

Subjects	Precision	Recall	F1
Sub1	0.86667	0.88361	0.8707
Sub2	0.88159	0.81884	0.83733
Sub3	0.97143	0.86151	0.91294
Sub4	0.95505	0.91613	0.93427
Sub5	0.71064	0.9133	0.79285
Sub6	0.60397	0.75129	0.66131
Average	0.83156	0.85745	0.8349

Table 6: Performance evaluation for visual-text model (Layer-3).

7 iPad Application Based on HDIC Framework

The proposed HDIC framework has been utilized in an iPad application developed for enhancing disaster situation reports and facilitating decision making processes. The implementation of the user interface (UI) is based on the officially supported tools for iOS design and coding, i.e., Apple’s Xcode 3 and its built-in Interface Builder and iOS Simulator applications. Figure 4 shows the main interface of the system, where users can browse the classified images associated with a specific situation report. Since it is not the focus of this paper, the details of the implementation are not introduced.



Federal Disaster Aid Approved For Missouri Flood Recovery

Figure 4: iPad application based on HDIC framework.

8 Conclusions and Future Work

In this paper, an hierarchical disaster image classification scheme based on MSDF and MCA is developed for enhancing disaster situation reports with relevant multimedia data and consequently improve the decision making process in disaster situations. The experimental results show the effectiveness of the proposed method. Furthermore, the proposed HDIC framework has been successfully in an iPad application for aiding EM personnel in disaster emergency response. However, there are several aspects of this algorithm to be improved. First, the hierarchical structure and weighting scheme are fixed for a specific scenario, where an adaptive approach is preferable. Second, the visual features are mainly low-level, and more mid-level features are needed to better describe the content of images. Finally, the range of disaster categories and subjects should be extended to serve more general purposes.

9 Acknowledgement

This material is based upon work supported by the U.S. Department of Homeland Security under grant Award Number 2010-ST-062-000039 and the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001.

References

[1] H. Abdi and D. Valentin. Multiple correspondence analysis. *Encyclopedia of measurement and statistics*, 2007.

[2] A. D. Amo and M. Farmer. Aided image understanding system. *Fuzzy Information Processing Society, NAFIPS*, pages 1–6, 2008.

[3] C. F. Barnes, S. Member, H. Fritz, and J. Yoo. Hurricane disaster assessments with image driven data mining in high resolution satellite imagery. *IEEE Transactions On Geoscience And Remote Sensing Symposium*, pages 1631–1640, 2007.

[4] H. Bayraktar and B. Bayram. Fuzzy logic analysis of flood disaster monitoring and assessment of damage in se anatolia turkey. *Recent Advances in Space Technologies*, pages 13–17, 2009.

[5] S. C. Chen, S. Sista, M. L. Shyu, and R. L. Kashyap. An indexing and searching structure for multimedia database systems. *SPIE Conference on Storage and Retrieval for Media Databases*, pages 262–270, 2000.

[6] S. S. Durbha, R. L. King, V. P. Shah, and N. H. Younan. Image information mining for coastal disaster management. *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 342–345, 2007.

[7] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[8] S. T. Gandhe, K. T. Talele, and A. G. Keskar. Image mining using wavelet transform. *Knowledge-Based Intelligent Information and Engineering Systems*, pages 797–803, 2007.

[9] C. C. Hsu and Z. Y. Hong. An intelligent typhoon damage prediction system from aerial photographs. *Knowledge-Based Intelligent Information and Engineering Systems*, pages 747–756, 2007.

[10] L. Lin and M. L. Shyu. Weighted association rule mining for video semantic detection. *International Journal of Multimedia Data Engineering and Management*, 1(1):37–54, Jan.-Mar. 2010.

[11] L. Lin, M. L. Shyu, G. Ravitz, and S. C. Chen. Video semantic concept detection via associative classification. *IEEE International Conference on Multimedia and Expo*, pages 418–421, Jul. 2009.

[12] G. Moser and S. B. Serpico. Classification of high resolution images based on mrf fusion and multiscale segmentation. *Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 277–280, 2008.

[13] A. Najab, I. Khan, and F. Ahmad. Principal component analysis based classification of settlements in satellite images. *Proceedings of the 6th International Conference on Frontiers of Information Technology*, 2009.