# Damage Assessment System For Image Data

Ezgi Naz Güngör*, Fatma Nisa Hopcu*, and Malek Baba*

*Hacettepe University, Ankara, Turkey

## Abstract

Damage assessment after disasters is an important task for governments and organizations, however it takes a lot of time to evaluate the damage especially if the disaster affected a large area. In this project we propose a solution for this problem by employing image classification methods on damaged structures. We gathered data from the web and categorized them to build the training dataset. We use deep convolutional neural network(CNN), support vector machine(SVM) and bag-of-visual-words(BoVW) models.

## 1. Introduction

A great number of natural disasters occur around the world every year. Detection of effects of a natural disaster is vital.

Damage assessment is important for detection of the disaster area and creating emergency rescue plans and other relief efforts according to the level of destruction after the disaster. Although warning systems can be used before the disasters, they're not always reliable.

There are several damage assessment systems(both pre-disaster and post-disaster) currently used by humanitarian organizations such as UNICEF but there aren't any automatic emergency disaster response systems. And most of this systems don't work with imagery data, instead they work with seismic activity  other signal-based data.These systems are mostly focused on earthquakes.

Post-disaster damage assessment systems are more useful for emergency disaster response and remote sensing systems. This systems can be developed using many image classification methods. In our project, we also used an image classifying approach, but using a regression approach is also possible.

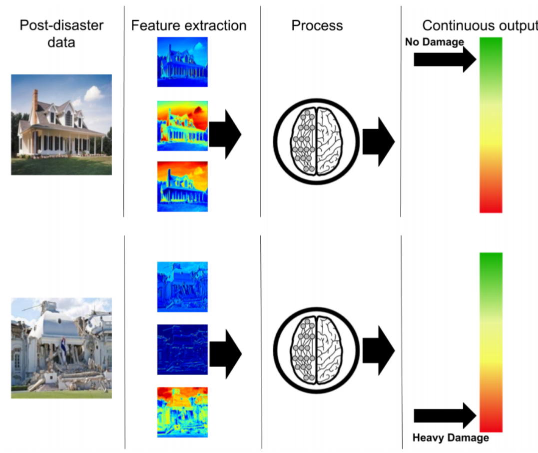
In this work, we propose a model to classify post-disaster imagery of buildings, roads and bridges into 9 pre-defined categories; none-to-little damage, mild damage and high damage for each structure. And used CNN, SVM, BoVW models to train our data.

This work is a first step the for a post-event damage assessment systems that can be reliable and used for organizing relief efforts and reduce the resources needed for the task, therefore significantly decrease the time needed to help people effected by disasters. Also decrease the efforts the recover from such disasters.

## 2. Related Work

Damage assessment is a topic often studied for disaster response where damage is estimated using satellite imagery, therefore using object identifications methods to identify buildings etc.However, in recent studies this topic is explored using data from social media, thus using non-aerial images. There're a few available data sets that can be used however they usually consist of images from specific disasters. So most studies use a combination of these disaster-specific data sets.In this study deep convolutional neural networks, either fine-tuned from pre-trained CNNs (ImageNet, AlexNet etc.) or the bag-of-visual-words model are used. (8)(2)(7)(12)

Although most studies on damage assessment use image classification methods, there are a few studies that predicts the damage in a continuous manner, where a regressor is placed at the end of the architecture to convert the output to a real value usually between 0 and 1(8). (Fig 1)




Regression Approach

In related studies, usage of data processing methods is very common since most images are crowded. Most images include people, scenery, sky, vehicles etc. By applying techniques such as color masking, it is possible to remove the unrelated parts from the images to get better prediction results.

Color Masking

In older studies, edge recognition was a popular method. In a study, by evaluating building edges and shadow edges, a collapsed building detection system was proposed for 1999 Gölcük earthquake. (11)

Most studies use a mix of existing datasets and manually-labelled datasets to work on. Existing datasets are usually composed of satellite imagery of natural disasters. In our project we use non-aerial images from web (Google, Bing etc.) and labeled them manually to create the dataset.


*Figure 1.* Images and their corresponding labels

## 3. The Approach

Since the problem that we are trying to solve, is a classification problem, we built our solution on two approaches: one is a statistics based approach and the other is a Deep learning based approach, then we compared our results on both of them. In The first approach, which is our baseline approach, We used Bag of visual words as feature extraction method, combined with a classifier. And in the second approach we used convolutional neural networks for an end-to-end learning of the image labels.

### 3.1. Bag Of Visual Words Model

The bag of words model in generally used in the natural language processing domain for textual document retrieval and classification. Using this model, each document is represented as an instance or a list of global important words and depending on the frequency of those words, the classification and the retrieval is done.

When the same model is applied to image data, instead of textual words, we consider visual words, which are in this case, patches of pixels that are global across the data set.

Building the model is done through the following steps

**(1) Feature extraction** : Interest points and their properties are extracted from the images as a set of global image descriptors. The goal of a descriptor is to provide a unique and robust description of an image feature. For this purpose we used ORB(9), however their exists, SURF(1)and SIFT(6)image descriptors. Which are in general better than edge descriptors.

SIFT stands for scale invariant feature transforms; which are set of translation,scaling, rotation invariant descriptors that describe key interesting points in the images.

SURF stands for speeded up robust features; which is similar to SIFT, however with a better performance in term of speed.

ORB stands for oriented fast and rotated brief; which is a combination of two feature extraction methods FAST and BRIEF. ORB is faster than both SURF and SIFT, while performing as well in many situations as they claimed in their paper.

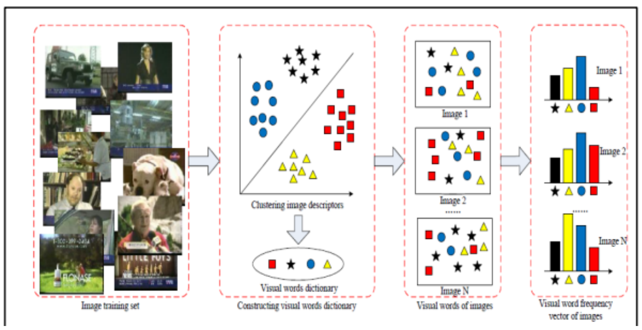
**(2) Visual words representation** : The idea is to group the feature descriptors of the all patches into clusters and the representatives of all resulting clusters are then used as the global visual words. For this purpose we used K-means clustering algorithm.

K-means algorithm uses the mean of a group of observations to determine the centre feature or point and use it as a centre of the resulted clusters. The number of clusters

K is predefined parameters.

**(3) Feature mapping** : After computing the visual words, each descriptor is linked to one visual word using the k-means algorithm.

**(4) Image representation** : After all the local features are mapped , an image can be globally represented by the frequency histogram of the visual words and the obtained histogram vector can be used later as the input for classification models like K-nearest neighbours or Support vector machines.




Bag of visual words

### 3.1.1. K-NEAREST NEIGHBOURS

K-nearest neighbour (KNN) algorithm is one of the most used classification algorithm due to it‘s simplicity. It can be used for regression as well. KNN algorithm is generative algorithm that try to learn the underlying representation of the train data to classify the images in the test set. It do that by calculating the distance between each image in the train set and the query image using a similarity function, generally euclidean distance is used for this purpose. After calculating the distance, k-most similar images are considered and their average label is used to to classify the query image.

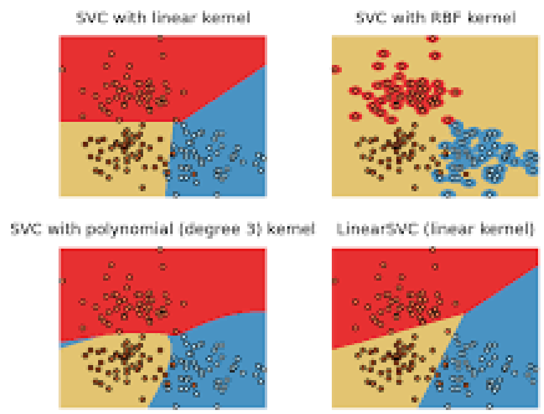
$$\text{Euclidean distance} = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2}$$

*Figure 2.* Euclidean distance

### 3.1.2. SUPPORT VECTOR MACHINES

Support vectors machines (SVM) is a popular classification algorithm, that can be used for regression also. SVM algorithm is a discriminative algorithm that try to find a hyperplane in an N-dimensional space(N — the number of features) that classifies the data points. This plane has the maximum margin, i.e the maximum distance between data points of both classes. By finding the best margin, it can ensure better accuracy for the future unseen data. In some cases the data is not linear separable, So non-linear kernel like (rbf or polynomial) instead of a linear kernel, can be used to map the features or the data points into a linearly separable space

SVM important parameters are Gamma G and Cost C. Those two parameters control the bias-variance trade off of the model. The cost parameter C determines the cost of the mis-classified input data, and the Gamma parameter determine the smoothness of the peaks in case of non-linear kernels.




SVM different kernels

## 3.2. Convolutional Neural Networks Model

Convolutions neural networks (CNN) have demonstrated very good performance in term of accuracy, when it comes to image classification task; due it is powerful feature representation capabilities. models like VGG16 (10) and ResNet-152 (4) have achieved very high Top-1 and Top-5 accuracy in the famous ImageNet(3) classification competition as we can see in Figure 1.

CNN Architecture transfer the input images into feature vectors, through learning different convolutions operations, that extract meaningful hierarchical features from input images. Those feature vectors are then fed to Fully Connected layers(NN), where the mapping to images labels are learned. Typically CNN networks have following operators, that are represented as layers:

**(1) Convolution Layer:**It convolves (which means taking the integral of the product of two function after one is reversed and shifted) the input image with a set of filters , which have parameters to be learned.

**(2) Pooling Layer:**It reduces the dimension of input, using the max element or the average.

**(3) Fully Connected Layer:** It learns non-linear representation from the last feature vector.

Learning the connection between the input images and the labels, is done though repetitive rounds of Forwards and backwards, where we feed our input images into the network and observe the loss, and in the Backward operations we update the weights of the networks depending on the value of the loss that we got. The backward phase is done through the famous backpropagation(5) algorithm.

Since training the models from scratch need a lot of time and data, we preferred to used Transfer learning strategy. Transfer learning help in transferring the knowledge learned in one domain and applying it to another domain. In image classification task generally the transfer learning performed on CNN models that had a good accuracy on the ImageNet image classification competition like VGGnet and ResNet. Transfer learning can be done in two ways and in our project we tested the both ways indeed. The first way is to continue training all the layers of the model, but this time in your own data set. The second method is to freeze the feature extraction layers and train just the final layers. Even though fine-tuning the whole model can lead to better results comparing to training just the final layers, in some case it can lead to outfitting due to the complexity of the generally used models like AlexNet and ResNet.

| Model | Size (M) | Top-1/top-5 error (%) | # layers | Model description |
|---|---|---|---|---|
| AlexNet | 238 | 41.00/18.00 | 8 | 5 conv + 3 fc layers |
| VGG-16 | 540 | 28.07/9.33 | 16 | 13 conv + 3 fc layers |
| VGG-19 | 560 | 27.30/9.00 | 19 | 16 conv + 3 fc layers |
| GoogleNet | 40 | 29.81/10.04 | 22 | 21 conv + 1 fc layers |
| ResNet-50 | 100 | 22.85/6.71 | 50 | 49 conv + 1 fc layers |
| ResNet-152 | 235 | 21.43/3.57 | 152 | 151 conv + 1 fc layers |

*Figure 3.* Different CNN architectures and their proprieties

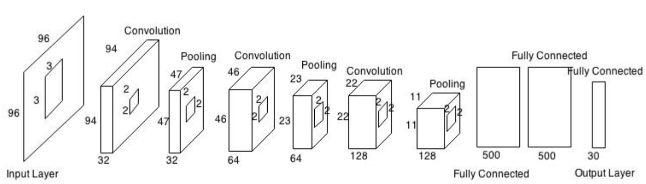



*Figure 4.* Simple CNN architecture

## 4. Experimental Results

In our project, we collect images from the web using Google and Bing search engines. Then, we labeled all images by hand to create our data sets. The number of images in our data sets that we initially created is as indicated in the table below. The first data set is divided into % 65 training, % 15 validation and % 20 testing. The second data set is divided into % 80 training, % 10 validation and % 10 testing for all the following experiments.

| First Dataset | | | | |
|---|---|---|---|---|
| Class | Low | Mild | High | Total |
| Building | 259 | 125 | 299 | 683 |
| Road | 205 | 142 | 128 | 475 |
| Bridge | 220 | 61 | 131 | 412 |
| | | | | 1570 |

| Second Dataset | | | | |
|---|---|---|---|---|
| Class | Low | Mild | High | Total |
| Building | 509 | 163 | 390 | 1062 |
| Road | 553 | 386 | 291 | 1230 |
| Bridge | 762 | 330 | 243 | 1335 |
| | | | | 3627 |

We used Bag of visual words as feature extraction method and we calculated accuracy, precision and recall values for each class separately.

Average accuracy value, total precision value and total recall value are respectively as follows % 85.016, % 84.93 and % 84.59 for our first data set.

Average accuracy value, total precision value and total recall value are respectively as follows % 14.345, % 12.209 and % 11.753 for our second data set. The results of the Bovw algorithm with KNN are shown below according to each class and data set.

| First Dataset | | | |
|---|---|---|---|
| **Class** | **Prec** | **Recall** | **Accuracy** |
| *Bridge_low* | 0.85 | 0.89 | 78.57 % |
| *Bridge_mid* | 0.73 | 0.78 | 89.36 % |
| *Bridge_high* | 0.95 | 0.74 | 74.19 % |
| *Building_low* | 0.86 | 0.86 | 86.89 % |
| *Building_mid* | 0.85 | 0.89 | 89.47 % |
| *Building_high* | 0.78 | 0.84 | 84.09 % |
| *Road_low* | 0.84 | 0.84 | 84.44 % |
| *Road_mid* | 0.83 | 0.89 | 89.65 % |
| *Road_high* | 0.91 | 0.84 | 84.61 % |

| Second Dataset | | | |
|---|---|---|---|
| **Class** | **Prec** | **Recall** | **Accuracy** |
| *Bridge_low* | 0.27 | 0.21 | 20.35 % |
| *Bridge_mid* | 0.10 | 0.11 | 11.53 % |
| *Bridge_high* | 0.08 | 0.07 | 9.52 % |
| *Building_low* | 0.23 | 0.19 | 23.15 % |
| *Building_mid* | 0.02 | 0.03 | 3.12 % |
| *Building_high* | 0.12 | 0.15 | 12.64 % |
| *Road_low* | 0.15 | 0.16 | 20.58 % |
| *Road_mid* | 0.08 | 0.09 | 13.04 % |
| *Road_high* | 0.03 | 0.04 | 12.28 % |

Also, we used Support Vector Machine instead of KNN. C parameter = 10, Kernel = ”Linear” are best chosen parameters for SVM classifier after doing a gird search over the space of the all parameters for the first dataset and kernel = ”bnf”, C = 10, G = 0.001 are the best parameters for the second dataset. Average accuracy value, total precision value and total recall value are respectively as follows % 70.016, % 78.93 and % 78.59 for our first data set.

Average accuracy value, total precision value and total recall value are respectively as follows % 25.345, % 15.209 and % 10.753 for our second data set

The growth of the data set increased both the complexity of the problem and the number of noise data. Because of this effect, accuracy value of the first data is higher than the accuracy of the second data set.

On the other hand, Convolutional Neural Networks yields well accuracy rate with imagery data set. Therefore, when classification of the our imaginary data sets, we chosen distinct pre-trained CNN architectures like AlexNet, VGG16 and ResNet50. Accuracy value of these CNN architectures are displayed below according to our data sets:

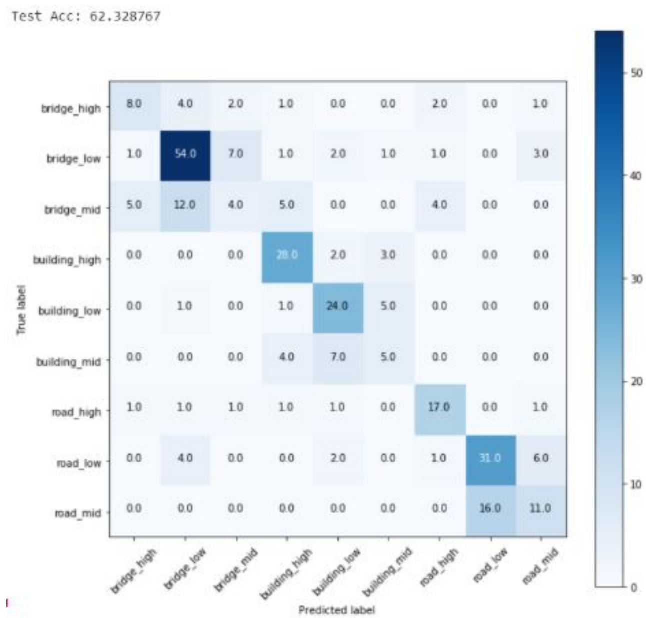
| First Dataset Accuracy | | | |
|---|---|---|---|
| Architecture | Train | Val | Test |
| AlexNet | 50.802 | 53.877 | 54.169 |
| VGG16 | 57.114 | 60.965 | 71.104 |
| ResNet50 | 61.924 | 56.579 | 65.524 |

| Second Dataset Accuracy | | | |
|---|---|---|---|
| Architecture | Train | Val | Test |
| AlexNet | 51.293 | 43.967 | 53.226 |
| Vgg16 | 50.364 | 52.212 | 58.064 |
| ResNet50 | 61.402 | 61.078 | 62.329 |

ResNet50 yields the best accuracy values compared to AlexNet and VGG16 for both data sets. Accuracy results of VGG16 architecture is better than the accuracy results of AlexNet architecture.

When we analyze the confusion matrix of the CNN result, we observed that ”Mild” classes of all structures are most problematic classes. After that, we ran the ResNet50 architecture without mild classes. The results of these process is like that: train accuracy is % 80.907, validation accuracy is % 74.638 and test accuracy is % 74.452 for the second data set.

Accuracy values of the second data set is lower than accuracy of the first data set as same as Bag Of Visual Words Algorithm. As we mentioned before, growing the data set cause also increasing the complexity of the problem and the number of noise data. In addition to these problems, data in-balancing is another problem to obtain good accuracy value.




Confusion Matrix

## 5. Conclusions

As we can see predicting the damage assessment level from general purpose images can be done using various image classification methods and sometimes regression methods, however due to the complexity and the richness of the problem, it requires the usage of a large dataset with enough representative images and more restrictions on the labeling side. Deep learning methods, specially CNN are the most suitable for this problem due to it‘s complexity.

## References

[1] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. Speeded-up robust features (surf), June 2008. ISSN 1077-3142. URL https://doi.org/10.1016/j.cviu.2007.09.014.

[2] Cooner, A., Shao, Y., and Campbell, J. Detection

of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 haiti earthquake, Oct 2016. ISSN 2072-4292. URL http://dx.doi.org/10.3390/rs8100868.

[3] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database, 2009.

[4] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

[5] Kelley, H. J. Gradient theory of optimal flight paths, 1960.

[6] Lowe, D. G. Distinctive image features from scale-invariant keypoints, November 2004. ISSN 0920-5691. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[7] Nguyen, D. T., Ofli, F., Imran, M., and Mitra, P. Damage assessment from social media imagery data during disasters, 2017. URL http://doi.acm.org/10.1145/3110025.3110109.

[8] Nia, K. R. and Mori, G. Building damage assessment using deep learning and ground-level image data, May 2017. ISSN null.

[9] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. Orb: An efficient alternative to sift or surf, 2011. URL https://doi.org/10.1109/ICCV.2011.6126544.

[10] Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2014.

[11] Turker, M. and San, B. T. Detection of collapsed buildings caused by the 1999 izmit, turkey earthquake through digital analysis of post-event aerial photographs, 2004. URL https://doi.org/10.1080/01431160410001709976.

[12] Vijayaraj, V., Bright, E. A., and Bhaduri, B. L. Rapid damage assessment from high resolution imagery, July 2008. ISSN 2153-7003.