Building Damage Assessment Using Deep Learning and Ground-Level Image Data

Karoon Rashedi Nia
School of Computing Science
Simon Fraser University
Burnaby, Canada
krashedi@sfu.ca

Greg Mori
School of Computing Science
Simon Fraser University
Burnaby, Canada
mori@cs.sfu.ca

***Abstract*—We propose a novel damage assessment deep model for buildings. Common damage assessment approaches utilize both pre-event and post-event data, which are not available in many cases. In this work, we focus on assessing damage to buildings using only post-disaster. We estimate severity of destruction via in a continuous fashion. Our model utilizes three different neural networks, one network for pre-processing the input data and two networks for extracting deep features from the input source. Combinations of these networks are distributed among three separate feature streams. A regressor summarizes the extracted features into a single continuous value denoting the destruction level. To evaluate the model, we collected a small dataset of ground-level image data of damaged buildings. Experimental results demonstrate that models taking advantage of hierarchical rich features outperform baseline methods.**

***Keywords*-Building Damage Assessment; Regression with Neural Networks; Hierarchical Models;**

I. Introduction

Numerous tragic natural disasters threaten vulnerable areas of the world each year. Although developing early warning systems is not always feasible using current technologies, it could be possible to utilize advanced computer vision approaches to manage post-event rescue and reconstruction plans efficiently. Computer vision based automatic damage assessment systems rely on optical data, which are processed to detect damaged areas and assess the level of destruction. According to the available data, these approaches either use only post-event data or both pre- and post-event images. In this work, we concentrate on the former.

Popular approaches that address automatic damage assessment use both pre-event and post-event aerial/satellite data to train classifiers (e.g. [1]). These algorithms classify damage into pre-defined categories. However, in our case we are interested in carrying out building damage assessment in a continuous fashion rather than classifying it, using only post-disaster data. To this end, we require a dataset of damaged buildings annotated with the level of damage, a feature extraction method to extract robust features out of post-disaster image data, and a regressor to summarize all the extracted features into a single continuous value representing the level of destruction. Figure 1 illustrates this idea.

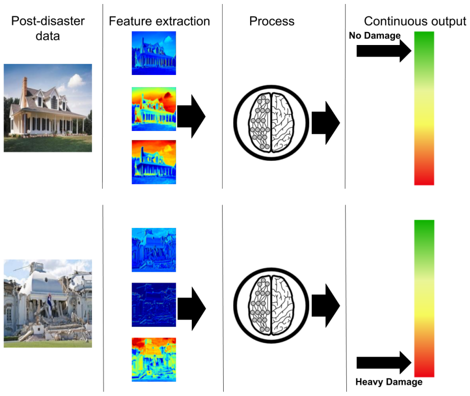


Figure 1. Given only post-disaster data we aim to output a continuous value representing the damage severity. First column represents the input to our algorithm, which is post-disaster images. The second column denotes extracted features which will be used by the regressor (the third column) to output a continuous damage level.

To our knowledge there is no publicly available dataset of ground-level images of areas affected by natural disasters and annotated with consistent labels. To deal with this problem, we collected a small dataset and annotated each instance with a label denoting the damage severity.

Inspired by the recent achievements of deep learning in different areas of computer-vision, we consider utilizing deep networks to extract robust features from the input source. We construct a model made up of three separate pipelines using different convolutional neural networks (ConvNets). Each pipeline is designed to perform a specific visual analysis. The first pipeline directly analyzes raw input images; however, the two other pipelines require a pre-processing step on the input data. The input data may contain irrelevant information to the context of our task, such as greenery, pedestrians, and cars, which affects the performance negatively. We consider extracting objects relevant to damage assessment by employing a semantic segmentation algorithm as a pre-processing step. Afterward, different ConvNets traverse the raw and pre-processed data and extract features. At last, the regressor is employed on the extracted features of the three streams and outputs a continuous value representing the level of damage.

The main contribution of this paper is the study of segmentation methods as an advantageous factor involved in damage detection and evaluation systems. In this work, we consider utilizing a semantic segmentation paradigm to improve the performance of the proposed algorithm. Beside directly analyzing post disaster data as the only input source to the algorithm, a semantic segmentation method is employed on these data to collect objects relevant to building damage assessment. Results indicate that semantically segmenting input data into objects of interest as foreground affects the performance positively.

II. Related Work

Damage detection and assessment is a well-studied topic and has been explored in multiple research areas. Solutions to computer vision problems, such as classification and segmentation, are the essential components of functional damage assessment systems.

A. Damage Detection Using Aerial/Satellite Imagery

Immediately responding to natural disasters plays a significant role in assisting the affected population. Due to the inaccessibility of many areas, satellite and aerial imagery are a valuable source of data for estimating the impact of a calamity. An example work is that of Gueguen et al. [1], who propose a semi-supervised framework to detect large-scale damage caused by catastrophes. They collected a dataset of 86 pairs of pre-event and post-event satellite imagery of impacted areas. For each pair, they extract features in 50×50 windows by using tree-of-shapes [2] as a descriptor of the shape. Based on these features the algorithm clusters areas, which then are used by human observers to obtain feedback on them. They use this feedback to train a linear Support Vector Machine (SVM) and classify the damaged areas. Thomas et al. [3] propose an automatic building damage assessment approach using pre-event and post-event aerial images. They consider an increase in the total number of edges appears on roof structures as a sign of damage. Yusuf et al. [4] evaluate the affected areas by calculating the difference between the brightness values in the pre and post earthquake satellite images.

B. Object Recognition Methods

Object recognition methods have also been brought to bear on the task of damage assessment. Extracting features using feature descriptors, such as SIFT [5] and HOG [6], and applying classifiers, such as SVM, have brought advancement to object recognition algorithms. However, during the past few years, the achievements of neural networks and the prevalence of large scale datasets (e.g. [7]) drew the attention of the computer vision community to new research directions. Construction of ConvNets follows conventional patterns, stacked convolutional layers followed by max-pooling and fully-connected layers. LeNet-5 [8], AlexNet [9], VGGNet [10], and ResNet [11] are examples along the progression towards more complex network structures following in this vein.

C. Semantic Segmentation

The goal of semantic segmentation is to classify each pixel of an image into predefined object / *stuff* classes. A large vein of work exists in this area. In our work, we consider utilizing a semantic segmentation algorithm to split the input data of our proposed algorithm into objects of interest and background. To this end, we use a state-of-the art network, DilatedNet [12], pre-trained on the ADE20K [13] dataset.

The ADE20K [13] dataset is collected for scene understanding and semantic segmentation, covering a diverse set of objects and scenes as well as providing detailed object annotations. Instances are annotated densely with object segments, their names, and object parts.

Recently, deep ConvNets have been used directly to predict pixel-level classification. For this purpose, the fully-connected layers which were used to output probability scores are replaced by convolution layers, allows ConvNets to preserve spatial dimensions. Fully Convolutional Networks (FCN) [14] up-samples intermediate extracted features from convolutional layers to preserve dimensions. DeepLab [15] and DilatedNet [12] are other examples of using deep ConvNets for dense prediction. They both use *atrous convolution* or *dilated convolution* in their fully convolutional network structures. The DilatedNet [12] replaces the *pool4* and *pool5* layers of the VGG-16 [10] with dilated convolution. DeepLab [15] utilizes *atrous convolution* with up-sampled filters, and fully-connected conditional random fields [16] in order to accurately segment along object boundaries.

III. Proposed Approach

Our goal in this work is to automatically assess damage to buildings caused by natural disasters. Common approaches classify damaged areas and detect changes by employing both pre-event and post-event aerial or satellite images to extract hand-crafted features as the input to linear classifiers such as SVM (e.g. [1]). In this work, we extract rich features from post-event images, as the only input source to our algorithm, and output a continuous value as a factor measuring damage severity rather than classifying damage into predefined categories.

We leverage multiple convolutional neural networks (ConvNets) to propose a novel deep model performing building damage assessment. Our model is made up of 3 feature streams; each represents attributes of the image data which are significant in damage assessment. Each pipeline comprises one or two ConvNets to extract rich features from the input data.

The color image feature stream employs a deep network (VGG [10]) to extract features from the raw input data (color

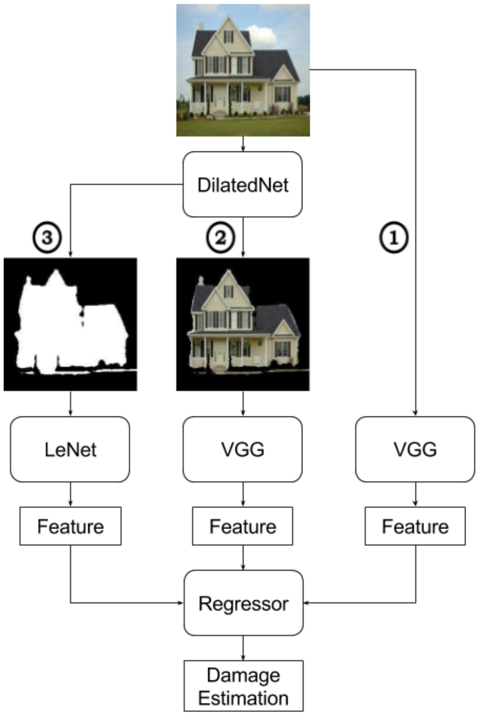


Figure 2. Overview of our proposed model. (1): Color image feature stream (Section III-A). A deep structure directly analyzes the input image data. (2): Color mask feature stream (Section III-B). A deep structure analyzes color masks of the image data. (3): Binary mask feature stream (Section III-C). A different deep structure is employed on the binary masks of the input data. The regressor utilizes extracted features to estimate the level of destruction.

image). The color mask feature stream applies a segmentation deep network (DilatedNet [12]) to semantically segment raw input data into objects of interest and then utilizes a deep model (VGG [10]) to obtain features out of the segmented images. The last pipeline, the binary mask feature stream, attains binary masks of the objects of interest in image data using the same segmentation method (DilatedNet [12]) and utilizes a different deep structure (LeNet [8]) to analyze their shape.

Having the extracted features, we could follow two directions. Using them to rank the instances (e.g. [17]) of damaged buildings based on the severity of damage, or summarize them into a single continuous value (regression) representing the level of destruction. In this work we perform regression using the extracted features. At last, the regressor exploits features extracted by the 3 streams to assess the level of destruction. An overview of the proposed model is illustrated in Fig. 2. In the following sections, we describe the behavior of each stream, their inputs, and corresponding outputs in detail. Moreover, we build several models based on these pipelines and evaluate their performance in Section IV.

A. Color Image Feature Stream

The color image feature stream is the first pipeline of our proposed model. This pipeline is designed to analyze raw input data as opposed to the two other pipelines which require a pre-processing step on the input data. The VGG network structure [10] extracts features from raw input data using the following procedure. Given an input image $X(Width \times Height \times Channels)$, which holds the raw pixel values of the image, several convolutional layers compute a dot product between their weights (W) and a small region of the input they are connected to (known as *receptive field*) and apply their bias offset (b) as shown in Equation 1. Afterward, an element-wise activation function (*ReLU*), shown in Equation 2, and a downsampling operation (known as $Pooling$) will be applied to the output of neurons.

$$z_j = f(\sum_i w_i x_i + b) \tag{1}$$

$$f(x) = max(0, x) \tag{2}$$

Features extracted by the convolutional layers are then processed by fully-connected layers, which hold neurons that are connected to all the neurons in the previous layer. The result is features of color images which then will be used by the regressor.

B. Color Mask Feature Stream

The second pipeline of our model is the color mask feature stream. Several visual factors, including camera position and angle, could affect the performance of vision-based systems by adding extra objects to the image, such as sky, trees, etc. These factors either slow down the learning process or have negative effects on the accuracy of vision-based algorithms. To address this issue in our work, we consider utilizing semantic segmentation algorithms to focus on the relevant regions in the input data rather than processing the whole image.

For this purpose, a deep structure (DilatedNet [12]) is used to segment images into objects of interest as foreground and the rest as background. Given an input image, DilatedNet collects all the instances of a pre-defined object set, which we consider as relevant, through an image and outputs those as foreground. A couple of output instances are shown in Figure 3.

Similar to Section III-A, several convolutional layers followed by an activation function (*ReLU*) and a pooling layer traverse color masks and extract features. These features are then used by fully-connected layers to provide the regressor (Section III-D) with features of color masks.

C. Binary Mask Feature Stream

Inspired by LeNet [8], which recognizes handwritten digits (essentially binary images) using a simple convolutional neural network, we consider learning the shape of intact

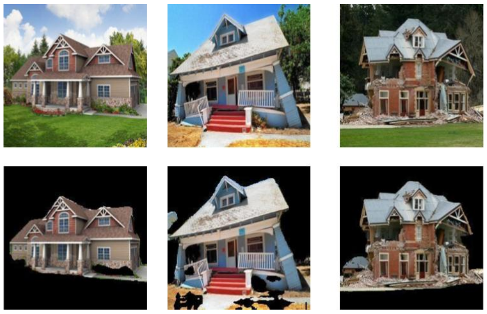


Figure 3. The first row: raw images which are used by the color image feature stream pipeline. Second row: corresponding color masks as the input to the color mask pipeline. Instances of greenery and the sky are omitted. Hence, the model is able to focus on relevant parts.

and destroyed buildings. The MNIST dataset [8], which has been used to train LeNet, contains black and white two-dimensional vectors representing handwritten digits. Using MNIST, LeNet successfully learned to distinguish between digits.

The third pipeline of our algorithm, the binary mask feature stream, is designed to learn the shape of damaged building parts. To this end, DilatedNet [12] is used to produce binary masks of our pre-defined object set which we expect to be relevant to damage assessment. The process is similar to Section III-B, however, the foreground (objects of interest) is marked as white, and the background is marked as black. In order to extract features corresponding to the shape of objects, the LeNet [8] network structure is then employed on the binary masks. Given an input $X(Width \times Height \times 1$ (input is binary so the dimension of channels is reduced to 1 as opposed to 3 in RGB images), we analyze it using a LeNet structure. Two convolutional layers (Equation1), each followed by a pooling layer which downsamples by a factor of 2, and a fully-connected layer using an activation function (*ReLU* in Equation 2), and another fully-connected layer, traverse the binary input and extract features to be used by the regressor (Section III-D).

D. Regression

The aforementioned pipelines provide hierarchical rich features in a feed-forward process. The end goal is to learn to compute a single continuous value out of these features. To this end, features obtained through the described streams are concatenated into a single feature vector. This vector is processed by a fully-connected layer, to infer a single value. A sigmoid function (Equation 3) is then used to map the output to a range between 0 and 1, which we expect to be the level of destruction.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Having the output makes the model able to utilize a loss function in conjunction with an optimization method in the error back-propagation process to update the weights in a way that the loss is minimized.

There is a difference between training a deep structure for classification and training it for regression. The loss function, which controls the learning process, in a regression network has to penalize outputs based on their distance to the ground truth. For example, for an unobserved data point equal to 2, a network which outputs 3 is performing better than a network which outputs 4, however, in a classification task both outputs are considered equally wrong. To address this issue in our work, we employ Euclidean distance (Equation 5) as the loss function and stochastic gradient descent (SGD) as the optimization method. Assuming $X_n = (x_1, ..., x_D)$, $Y_n = (y_1, ..., y_D)$, and $Z_n = (z_1, ..., z_{D'})$ as the extracted features of the n_{th} sample through the color image, color mask, and binary mask pipelines, we can formulate the process as the following optimization problem:

$$p_n = \phi(X_n^\frown Y_n^\frown Z_n) \tag{4}$$

$$\underset{w}{\text{argmin}} \quad \frac{1}{2N} \sum_{i=1}^{N} \| t_n - S(p_n) \|_2^2 \tag{5}$$

where $^\frown$ stands for concatenation. ϕ is the function of the fully-connected layer applied to the extracted features, N is the total number of training samples, t_n is the ground truth on example n, $S(p_n)$ denotes applying Sigmoid function (Equation 3) to the predicted label for the sample n, and w stands for all the learnable weights in convolutional and fully-connected layers.

IV. Experiments

A. Dataset

We collected a novel dataset of ground-level post-disaster images of buildings affected by natural disasters. All the images are obtained through the internet using three sources: Virtual Disaster Viewer [18], Open Images Dataset [19] and Google image search engine. We collected and refined images and labels through the different stages. There are 200 images in the final training set and 50 images in the final testing set [1]. The collected images had different sizes. Due to memory and space considerations we downsampled all the instances to 224×224 to be used by color image and color mask pipelines and 28×28 to be used by binary mask pipeline. Moreover, by downsampling, we were able to take advantage of pretrained state of the art models which use the same image size.

[1] The dataset can be downloaded at: http://vml.cs.sfu.ca/wp-content/uploads/building/buildingdataset.tar.gz

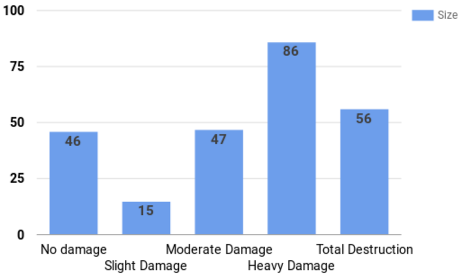


Figure 4. Distribution of five damage levels over the collected dataset.

B. Data Annotation

In order to pair each image with the damage estimation, we utilized the damage classification scheme described in the damage assessment report of the Haiti earthquake [20] to annotate each image in the training set with a value between 0 and 1 at a step size of 0.25. Greater values imply more destruction.

The damage assessment report [20] classifies damage to buildings of Haiti, caused by the earthquake in 2010, based on visual attributes of walls, floors, and roofs. Inspired by their approach, we manually classified images of the dataset into five levels. No damage, slight damage, moderate damage, heavy damage, and total destruction levels are described with 0, 0.25, 0.5, 0.75, and 1 respectively. Figure 4 illustrates the distribution of labels over the dataset.

C. Training Settings and Results

We acquired color masks and binary masks of objects of interest using a DilatedNet [12] pretrained on the ADE20K [13] dataset. We defined a set of 29 out of 900 objects of ADE20K that we consider as relevant to building damage assessment, including wall, building, and ceiling [2]. DilatedNet is fed by raw images and based on pixel-wise classifications, and collects instances of objects of our interest. As the first step of training, we converted color images, color masks, and binary masks as well as labels to HDF5 data models to be used by Caffe [21] as the input source. Each image in the data model is paired with its label. These labels are used in the backward propagation of errors (backpropagation) to compute the gradient of the Euclidean loss function with respect to all the weights. Stochastic Gradient Descent (SGD) uses the computed gradient to update the weights and minimize the Euclidean loss.

We built several models using described pipelines and evaluated them on our test set. Results are provided in Table I. Results denote the color mask feature stream contributes more than other pipelines to the performance of the system.

Table I

EVALUATION OF PROPOSED MODELS BASED ON EUCLIDEAN DISTANCE. LOWER VALUES REPRESENT BETTER PERFORMANCE. COLOR IMAGE AND COLOR MASK PIPELINES UTILIZE VGG [10] PRETRAINED ON IMAGENET [22] AND FINE-TUNED ON OUR COLOR IMAGE AND COLOR MASK DATASETS RESPECTIVELY. THE BINARY MASK PIPELINE EMPLOYS LENET [8] PRETRAINED ON THE MNIST DATASET AND FINE-TUNED ON OUR BINARY MASK DATASET.

Model components			**Euclidean distance**
Color image feature stream	**Color mask feature stream**	**Binary mask feature stream**	
✔	—	—	0.20
—	✔	—	0.18
—	—	✔	0.33
✔	✔	—	**0.17**
✔	—	✔	0.20
—	✔	✔	0.19
✔	✔	✔	**0.18**

Table II

TRAINING SETTINGS. “LR” STANDS FOR LEARNING-RATE AND “FC” STANDS FOR FULLY-CONNECTED LAYER. “BASE LR” APPLIES TO ALL THE LAYERS, EXCEPT THE LAST FC LAYER. SETTINGS ARE RANKED RELATIVELY BASED ON THEIR LR.

lr **Setting**	**base lr**	**last fc weights learning rate**	**last fc bias term lr**
high lr	0.0001	0.001	0.002
medium high lr	0.0001	0.0005	0.001
medium lr	0.0001	0.0002	0.0005
low lr	0.0001	0.0001	0.0002

Moreover, in order to realize whether changing the training strategies affects the performance, we chose our two best performing models in Table I (color image + color mask as the best-performing model, and color image + color mask + binary mask as the second best-performing model) and conducted further experiments on them. We modified the hyper-parameters we used to train our regressor by defining new learning rates. These settings are shown in Table II. The results of applying different training settings on the two best-performing models are shown in Table III. These models reached their best results using different settings. Samples of testing input and output of the best model (color image + color mask using medium learning-rate) are shown in Figure 5. Moreover, examples of failure cases of this algorithm are illustrated in Figure 6.

V. CONCLUSION

In this work, we presented a novel hierarchical model performing building damage assessment in a continuous fashion using post-disaster images of damaged buildings as the input. The model contains three different pipelines and each pipeline is made up of one or two different ConvNets. These pipelines are designed to extract features from the

[2] Wall, building, floor, ceiling, road, windowpane, sidewalk, earth, door, house, field, rock, column, skyscraper, path, stairs, runway, screendoor, stairway, bridge, countertop, hovel, awning, booth, pole, land, banister, escalator, and tent are considered as relevant.

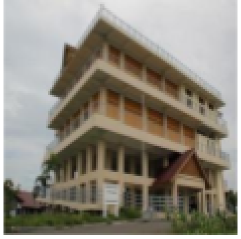
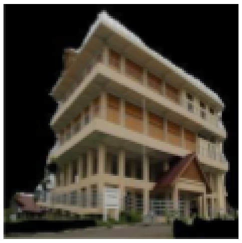
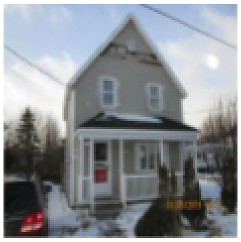
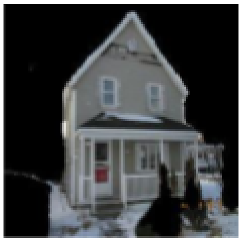
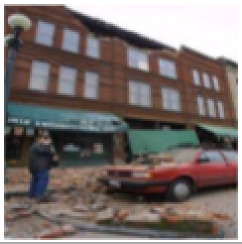
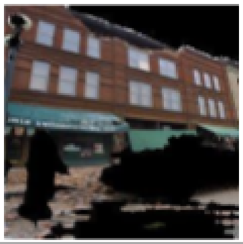
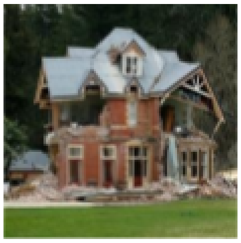
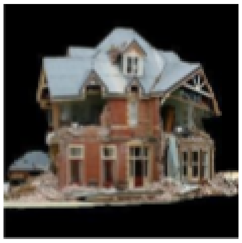
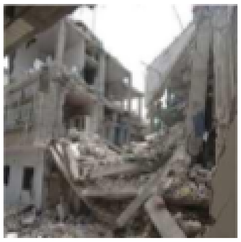
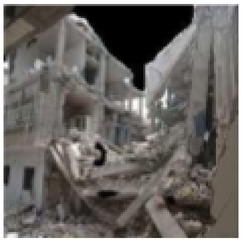
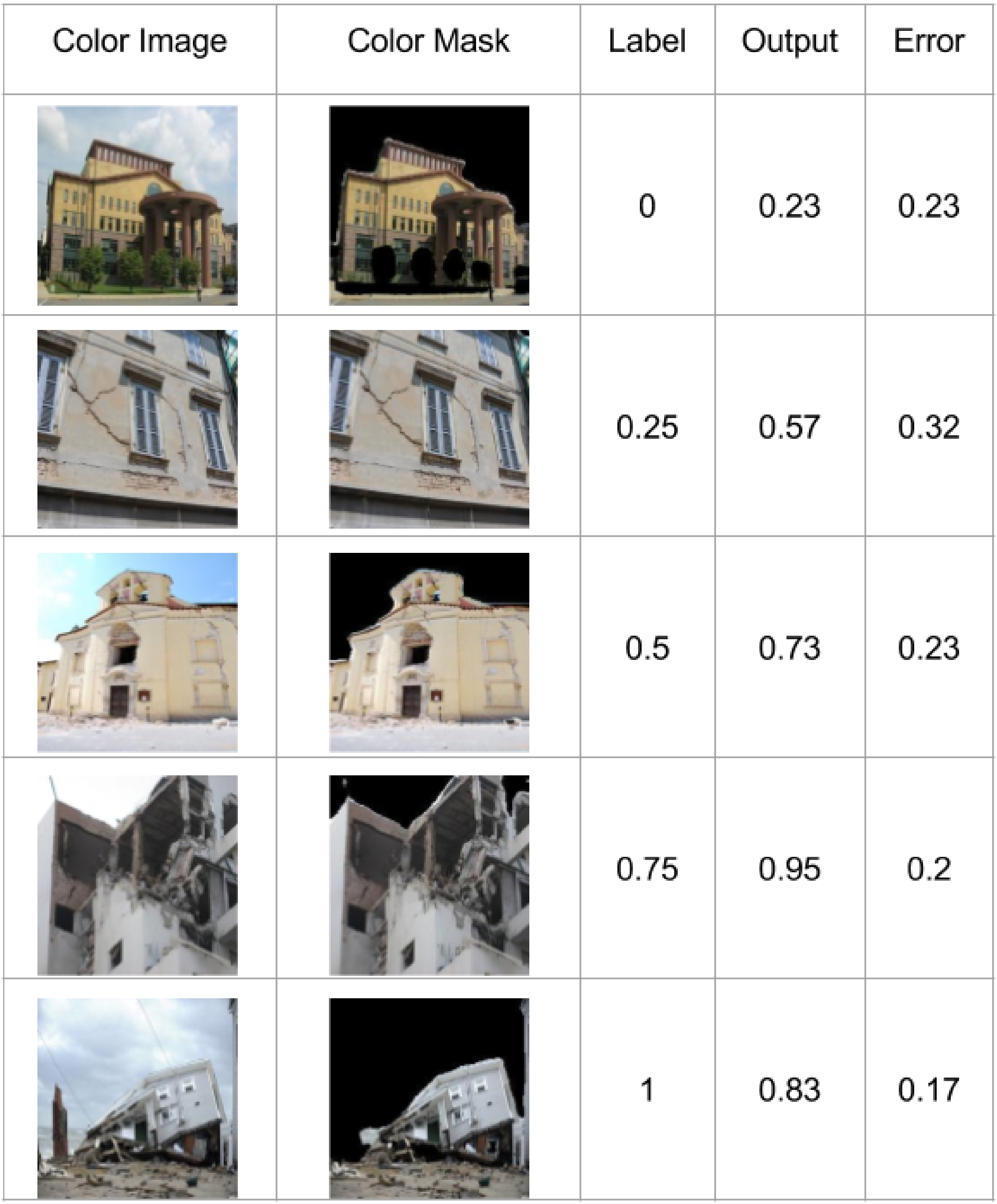
Color Image	Color Mask	Label	Output
		0	0.03
		0.25	0.30
		0.5	0.54
		0.75	0.74
		1	0.97

Figure 5. Samples of testing input and output of our best-performing model, color image + color mask feature streams trained by medium lr (Table III). Left two columns are inputs to the algorithm, the third column is ground-truth, and the last column is the estimated damage level.

raw input data, objects that we consider as relevant to damage assessment, and the shape of intact and damaged building parts. Additionally, we collected a dataset of post-disaster images, to train and evaluate our models. We built several models using combinations of the three pipelines and evaluated their performance. The results show that models taking advantage of features of raw input data and color mask of relevant objects to building damage assessment have



Color Image	Color Mask	Label	Output	Error
		0	0.23	0.23
		0.25	0.57	0.32
		0.5	0.73	0.23
		0.75	0.95	0.2
		1	0.83	0.17

Figure 6. Examples of failure cases of our best-performing model. Left two columns are inputs to the algorithm, the third column is ground-truth, the fourth column is the estimated damage level, and the last column is the error (Euclidean distance).

potential to perform better than other models.

References

[1] L. Gueguen and R. Hamid, "Large-scale damage detection using satellite imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1321–1328.

[2] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 860–872, 2000.

Table III
Results of further experiments conducted on our two best-performing models, pretrained VGG on ImageNet and fine-tuned on color image and color mask training sets as the best-performing model and its combination with pretrained LeNet on MNIST and fine-tuned on binary mask training set as the second best-performing model. We trained them using different learning rates described in Table II.

Model **Setting**	**color image + color mask feature streams**	**color image + color mask + binary mask feature streams**
High lr	0.172	0.177
Medium high lr	0.169	0.176
Medium lr	**0.166**	0.175
Lowh lr	0.172	**0.173**

[3] J. T. A. K. K. Bowyer, “Towards a robust, automated hurricane damage assessment from high-resolution images,” in *International Conference on Web Engineering (ICWE)*, 2011.

[4] Y. Yusuf, M. Matsuoka, and F. Yamazaki, “Damage assessment after 2001 gujarat earthquake using landsat-7 satellite images,” *Journal of the Indian Society of Remote Sensing*, vol. 29, no. 1-2, pp. 17–22, 2001.

[5] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[9] A. K. I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Neural Information Processing Systems (NIPS)*, 2012.

[10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.

[13] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ade20k dataset,” *arXiv preprint arXiv:1608.05442*, 2016.

[14] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *arXiv preprint arXiv:1606.00915*, 2016.

[16] V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” *Adv. Neural Inf. Process. Syst*, vol. 2, no. 3, p. 4, 2011.

[17] K. Crammer and Y. Singer, “Pranking with ranking,” in *Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 641–647.

[18] *Virtual Disaster Viewer*. Retrieved April, 2016, from http://vdv.mceer.buffalo.edu/vdv/select_event.php.

[19] *Introducing the Open Images Dataset*. Retrieved September, 2016, from https://research.googleblog.com /2016/09/introducing-open-images-dataset.html.

[20] *Building Damage Assessment Report*. Retrieved April, 2016, from http://www.eqclearinghouse.org/co/20100112-haiti/wp-content/uploads/2010/02/PDNA_damage_assessment_report_v03-1.pdf.

[21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *Computer Vision and Pattern Recognition (CVPR)*, 2009.