EUROPEAN MECHANICAL SCIENCE

RESEARCH ARTICLE

e-ISSN: 2587-1110

Assessing household damages using multi-model deep learning pipeline

Fatih Kıyıkçı[1], H. Onur Cunedioğlu[1], Enes Koşar[1], M. Eren Bekin[1], Fatih Abut[2*], M. Fatih Akay[2]

[1]Anadolu Sigorta, Istanbul, Turkey.
[2]Çukurova University, Department of Computer Engineering, Adana, Turkey

Orcid: F. Kıyıkçı (0000-0003-3949-5680), H.O. Cunedioğlu (0000-0002-4782-1768), E. Koşar (0000-0001-9757-2483), M.E. Bekin (0000-0002-9024-250X), F. Abut (0000-0001-5876-4116), M.F. Akay (0000-0003-0780-0679)

Abstract: Since the beginning of the pandemic, the home insurance sector has suffered from various difficulties. One of the most important difficulties was assessing the damages in the insurance owners' homes. Due to the current pandemic, letting the experts assess the damages in place is a life-threatening risk. Therefore, the idea of automatically assessing the damage is born. This study aims to create a full report for home damages using Convolutional Neural Network (CNN) and various large deep learning architectures such as EfficientNet, ResNet, U-Net, or Feature Pyramid Network (FPN). Multiple models for tasks such as binary classification and instance segmentation were developed to create an end-to-end reporting pipeline. In more detail, the pipeline consists of two binary classification models and a segmentation model. Binary classification models are responsible for detecting if the picture is indoors and if there is a wall in the picture, whereas the instance segmentation model is responsible for segmenting the damaged parts of the wall class. The effectiveness of the pipeline was measured using different metrics for each task, including accuracy, F1, Intersection over Union (IoU), and Dice scores. The data for each task is labeled by hand and fed to models. The results show that the constructed pipeline can successfully classify and segment the given images according to the needs of our project. The project will affect the home insurance assessment procedure and time spent tremendously by automatizing these repetitive processes.

Keywords: Deep learning, convolutional neural networks, semantic segmentation, classification.

1. Introduction

Since the beginning of the pandemic, people tend to make less contact with strangers or even with relatives as they should be. However, in particular areas of the natural flow of life, getting in touch with somebody is inevitable, i.e., during shopping, working, and public transporting. The pandemic has changed the way people live drastically. During this era, people are witnessing how the most deep-seated habits, traditions, and the way people behave are transforming at a killing pace [1].

With the advanced computer technologies and the pandemic force, contactless applications are trending as never been before, and it seems that they will still be the hot topic for a while [2-4]. Before the pandemic, all these applications were in the "Nice-To-Have" category, but now, they have settled in the "Must-Have" shelf robustly. Considering these new pandemic normals, the "Household Damage Assessing" project has been implemented to automate these cumbersome processes.

Several studies have been conducted in the literature to assess household damages using different methods. For example, Perez et al. [5] used close-up images to classify the defect in the building based on VGG-16. The study predicts the defect and localizes the possible defect using Class Activation Mapping (CAM). Yu et al. [6] proposed a novel method based on deep convolutional neural networks to identify and localize damages of building structures equipped with smart control devices. Li et al. [7] designed a platform to automatically estimate home damage levels in disaster areas through crowdsourcing ideas and CNN approaches. Feng et al. [8] developed a highly accurate structural damage detection method using a deep convolutional neural network with transfer learning. Naito et al. [9] proposed a method for detecting damaged buildings in the event of an earthquake using machine learning models and aerial photographs. These previous solutions, techniques, or discussions do not offer an industry-level solution. Instead, the topic has been covered by an academic view, and the images studied are not from real-world data, i.e., not shot on mobile phones or taken

* Corresponding author.
Email: fabut@cu.edu.tr

European Mechanical Science (2022), 6(2): 138-142
https://doi.org/10.26701/ems.1031595
Received: December 5, 2021 — Accepted: March 1, 2022

by building surveyors.

This study aims to create a full report for home damages using CNN and various large deep learning model architectures such as EfficientNet, ResNet, U-Net, or Feature Pyramid Network (FPN). Mainly, the contributions of this study can be summarized as follows:

- We created the input data based on 3.700 labeled ground truth images retrieved from Anadolu Sigorta's own database. Those images are only from house damage claims.
- Multiple models for binary classification and instance segmentation were developed to create an end-to-end reporting pipeline. The pipeline consists of two binary classifiers and a segmentation model in more detail. Binary classifiers are responsible for detecting if the picture is indoors and if there is a wall in the picture. In contrast, the instance segmentation model is responsible for segmenting the damaged parts of the wall class.
- The performance of the classification models has been evaluated by calculating the confusion matrix and the accuracy, F1, Intersection over Union (IoU), and Dice scores.

The rest of the paper is structured as follows. Section 2 gives information about the dataset and methodology. Section 3 presents the results and discussion. Section 4 concludes the paper along with possible research directions.

2. Dataset and Methodology

The model pipeline, illustrated in Figure 1, is constructed by three different deep learning models connected with their image inputs and filtered outputs in an end-to-end fashion.

The input pictures from a home insurance claim are first fed to a binary classifier which classifies the picture as indoor or notIndoor. Then, if an indoor image is detected, another binary classifier classifies the image as wall or notWall. Finally, suppose an image of a wall is detected. In that case, it is fed to the damage segmentation model, which segments the damages that are present in the image with respect to their damage class and produces output mask images.

The utilized input data consists of images retrieved from Anadolu Sigorta's own database and are only from house damage claims. Figure 2 shows some samples from each pipeline class. The data can be tagged in a wide variety of semantic classes. We have used and labeled the data in three different semantic meanings for their respective objectives: indoor-outdoor classification, wall-not wall classification, and semantic segmentation of wall damages with four different classes of wall damages: crack, water stain, paint peeling, and shatter. We used 1.425 labeled

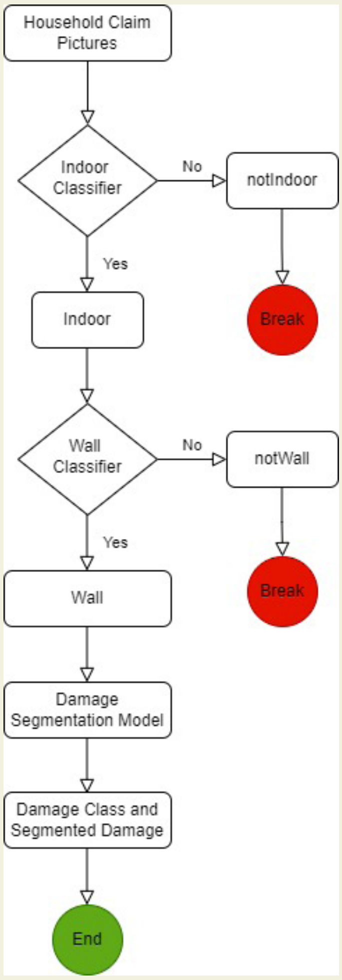

Figure 1. Flowchart of the home damage assessment pipeline

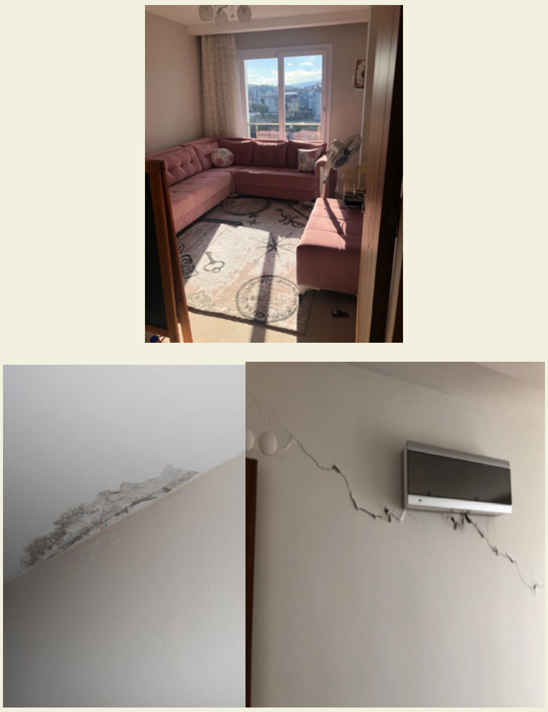

Figure 2. Samples from each pipeline class (indoor, wall, and crack)

 https://doi.org/10.26701/ems.1031595

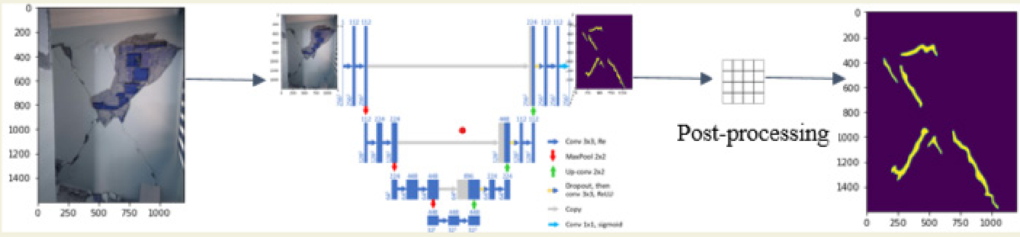

Figure 3. Visualization of segmentation process of a random sample using U-Net model architecture

images for each classification task and 2.374 labeled images for the segmentation task.

For all objectives, we have used a variety of pre-processing methods which go hand in hand with both the objective and used modeling methods, such as image augmentation (i.e., flip, blur, or saturation), resizing, or normalizing the images. All images in all objectives have been resized to a 384x384 resolution to standardize the image resolutions as input to the deep learning model, reducing the training time and required VRAM.

For classification objectives, we haven't used any pre-processing methods other than resizing and normalizing. For the segmentation task, we have used Gaussian blurring and random brightness contrast augmentation function from the "Albumentations" [10] library in Python, with the probability of application p=0.3 and a custom random box-blurring augmentation, which blurs a small box-shaped portion of the image with the probability of application p=0.7.

Several metrics have been proposed in the literature to evaluate the performance of binary classification and segmentation models. We summarize the most popular metrics for assessing the accuracy of classification and segmentation models.

- **Accuracy / F1-score** can be defined for a bi-class problem or for each class in the case of multiclass as follows (TP, TN, FP, and FN refer to the true positives, true negatives, false positives, and false negatives, respectively):

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \#(1)$$

$$F1-score = \frac{2*TP}{2*TP+FP+FN} \#(2)$$

- **Intersection over Union (IoU)**: IoU is defined as the area of intersection between the predicted segmentation map A and the ground truth map B, divided by the area of the union between the two maps:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP+FP+FN} \#(3)$$

- **Dice:** Dice coefficient is a popular metric for image segmentation, which can be defined as twice the overlap area of predicted and ground-truth maps, divided by the total number of pixels in both images.

$$Dice = \frac{|A \cap B|}{|A|+|B|} = \frac{2*TP}{2*TP+FP+FN} \#(4)$$

We used CNN along with various large deep learning model architectures. CNN has become hugely popular because of its architecture which automatically performs feature extraction for image classification. CNN uses two operations called 'convolution' and 'pooling' to reduce an image into its essential features and uses those features to understand and classify the image.

We experimented with a wide range of combinations for training the deep learning models for each of their respective tasks and evaluated each experiment using the evaluation metrics defined in Section 2.3. We used the PyTorch [11] library in Python for experimenting and constructing deep learning models. We utilized the transfer learning method, which transfers weights of models trained on large datasets, such as ImageNet, to a new model that we train on our specific objective and data. We used pre-trained models from timm [12] and SMP [13] packages for classification and segmentation, respectively. Furthermore, we used CrossEntropyLoss from the torch library as the main loss function since the data is not in an imbalanced state.

For the segmentation task, we have experimented with a large family of deep learning model architectures such as EfficientNet [14] or ResNet [15] overall objectives. After that, as illustrated in Figure 3, we also experimented with U-Net [16] and FPN [17] architectures for the segmentation task, using the SMP library, which serves segmentation models that are using pre-trained EfficientNet or ResNet architecture weights. Figure 4 represents some damage segmentation model outputs overlaid on predicted images.

Also, we constructed a custom learning rate scheduler dependent on the number of epochs and Adam optimization function from the torch library. We used 20% of the training dataset as validation and used two different test datasets. After the training phase, we applied different post-processing methods to the output for stabilizing the predictions and reducing the number of errors.

We used thresholding to reduce the number of incorrectly predicted pixels while keeping the number of correctly predicted pixels roughly the same. Thus, we have selected different thresholds and experimented with optimizing the evaluation metrics. We have found that the value 0.2 was the optimal threshold value for all four classes.

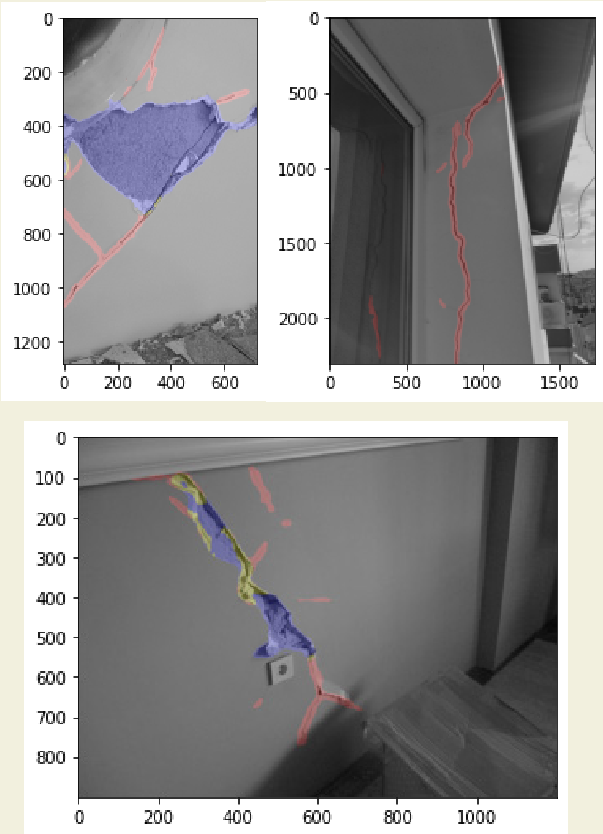

Figure 4. Damage segmentation model outputs overlaid on predicted images

To reduce the number of false positives (i.e., ground truth has no damage, the prediction has little damage), we have used a min_pixel_ratio value, which returns an empty mask if the predicted number of pixels over total pixels in the image is not greater than the min_pixel_ratio. This approach prevents the output of 1 pixel for an empty mask, resulting in a score of 0. According to our experiments, this value is optimal at 2%.

3. Results and Discussion

Table 1 and Table 2 show the experiment and confusion matrix results for binary indoor classification models, respectively. Similarly, Table 3 and Table 4 show the experiment and confusion matrix results for binary wall classification models, respectively.

The best indoor/notIndoor model in our experiments has achieved an accuracy of 0.885 and an F1-score of 0.878, considering that the dataset is created using real-life claims data and has many uninformative images. In this case, the model has encountered only 41 errors from 357 images, although only 855 images have been used for training (excluding validation).

Wall classification and indoor classification are very similar problems in terms of their technical objective and complexity. The best wall classifier, which was close to the performance exhibited by the indoor classifier, yielded an accuracy of 0.879 and an F1-score of 0.877 and had 43 errors in 357 photographs by using 855 images for training.

Table 1. Experiment results for binary indoor classification models

Backbone name	Accuracy	F1-score
EfficientNet_B0	0.874	0.862
EfficientNet_B1	0.885	0.878
EfficientNet_B2	0.868	0.860
ResNet18	0.865	0.852
ResNet50	0.865	0.848

Table 2. Confusion matrix results of predictions from EfficientNet-B1 model for indoor classification

	notIndoor	Indoor
notIndoor	114	20
Indoor	21	201

Table 3. Experiment results for binary wall classification models

Backbone name	Accuracy	F1-score
EfficientNet_B0	0.860	0.857
EfficientNet_B1	0.871	0.869
EfficientNet_B2	0.879	0.877
ResNet18	0.831	0.830
ResNet50	0.750	0.750

Table 4. Confusion matrix results of predictions from EfficientNet-B2 model for wall classification

	notWall	Wall
notWall	179	25
Wall	18	134

The segmentation models were trained using 1.401 images and tested using 973 images in two test sets. Table 5 shows the experiment results of models on both test sets (averaged per sample). The best damage segmentation model is an FPN architecture model using an EfficientNet-B2 as a backbone, which has reached a Dice score of 0.647 and an IoU score of 0.576.

Table 5. Experiment results for the damage segmentation models

Backbone name	Architecture	Dice Score	IoU Score
EfficientNet_B0	FPN	0.644	0.573
EfficientNet_B1	FPN	0.641	0.570
EfficientNet_B2	FPN	0.647	0.576
EfficientNet_B0	U-Net	0.632	0.560
EfficientNet_B1	U-Net	0.634	0.562
EfficientNet_B2	U-Net	0.636	0.565

4. Conclusion and Future Work

This study proposed models for automatically assessing household damages using multi-model deep learning. In more detail, two binary classification models and a segmentation model have been developed based on a dataset created from the database of Anadolu Sigorta. Binary classification models are responsible for detecting if the picture is indoors and if there is a wall in the picture. In contrast, the instance segmentation model is responsible for segmenting the damaged parts of the wall class. Experiments have been made using transfer learning with different backbone architectures and pre-trained weights. A pipeline solution with three steps has been implemented for the project requirements. For all tasks, several experiments have been conducted with different model architectures and backbones. At the final state, the pipeline consists of one EfficientNet-B1 indoor classification model, one EfficientNet-B2 wall classification model, and one FPN damage segmentation model with EfficientNet-B2 backbone.

Despite having a limited dataset, the model pipeline achieved relatively good scores and stable results on all three objectives. We believe that with time, as we label more images in an accurate and standardized fashion, the results will get better. In future work, we plan to expand the limits of the damage detection pipeline to not only detect and segment the wall damages but also do the same for tile and parquet damages and predict their dimensions while creating a detailed report of damage for the claim. Other possible future research direction includes automatically categorizing and assessing automobile damages from road traffic accidents.

5. References

[1] Di Crosta, A., Ceccato, I., Marchetti, D., La Malva, P., Maiella, R., Cannito, L., Di Domenico, A. (2021). Psychological factors and consumer behavior during the COVID-19 pandemic". PloS one, 16(8), e0256095.

[2] Puriwat, W., Tripopsakul, S. (2021). Explaining an adoption and continuance intention to use contactless payment technologies: during the COVID 19 pandemic. Emerging Science Journal, 5(1): 85-95.

[3] Al Tarabsheh, A., Yaghi, M., Younis, A., Sarker, R., Moussa, S., Eldigair, Y., Hajjdiab, H., El-Baz, A., Ghazal, M. (2021). Towards Contactless Learning Activities during Pandemics Using Autonomous Service Robots. Applied Sciences, 11, 10449.

[4] Li, X., Yuen, K. F., Wang, X., Wong, Y. D. (2021). Contactless technologies adoption during the coronavirus pandemic: a combined technology acceptance and health belief perspective. Technology Analysis & Strategic Management. doi: 10.1080/09537325.-2021.1988919.

[5] Perez, H., Tah, J. H., Mosavi, A. (2019). Deep learning for detecting building defects using convolutional neural networks. Sensors, 9(16), 3556.

[6] Yu, Y., Wang, C., Gu, X., Li, J. (2019). A novel deep learning-based method for damage identification of smart building structures. Structural Health Monitoring, 18(1): 143-163.

[7] Li, Z., Tian, K., Wang, F., Zheng, X., Wang, F. (2016). Home damage estimation after disasters using crowdsourcing ideas and Convolutional Neural Networks. In: 2016 5th International Conference on Measurement, Instrumentation and Automation.

[8] Feng, C., Zhang, H., Wang, S., Li, Y., Wang, H., Yan, F. (2019). Structural damage detection using deep convolutional neural network and transfer learning. KSCE Journal of Civil Engineering, 23(10): 4493-4502.

[9] Naito, S., Tomozawa, H., Mori, Y., Nagata, T., Monma, N., Nakamura, H., Shoji, G. (2020). Building-damage detection method based on machine learning utilizing aerial photographs of the Kumamoto earthquake. Earthquake Spectra, 36(3): 1166-1187.

[10] Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. Information, 11(2): 125.

[11] Paszke, A. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems, 32: 8024-8035.

[12] Wightman, R. (2019) PyTorch Image Models. GitHub. doi:10.5281/zenodo.4414861

[13] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[14] Tan, M., Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, 6105-6114.

[15] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: Proc. of the IEEE conference on computer vision and pattern recognition, 770-778.

[16] Ronneberger, O., Fischer, P., Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention, 234-241.

[17] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. (2017). Feature pyramid networks for object detection. In: Proc. of the IEEE conference on computer vision and pattern recognition, 2117-2125.