IMAGE UNDERSTANDING FOR PROPERTY CLAIM SETTLEMENT: STATUS AND CHALLENGES

Dr. Andrey Lutich*
sachcontrol GmbH

Dresden, February 2019

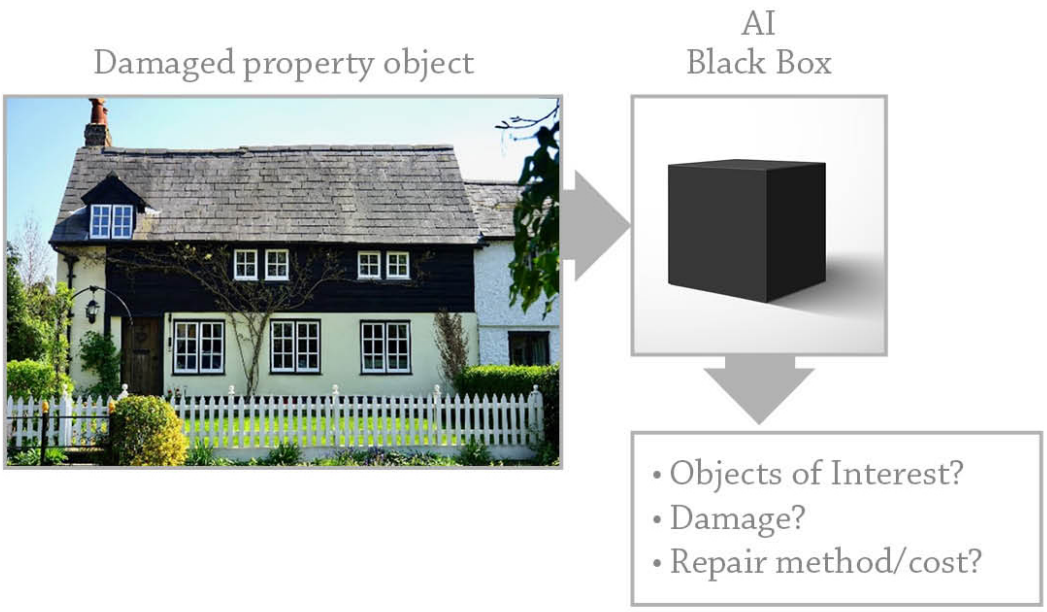
The insurance claim settlement process is in the heart of the relations between an Insurer and an Insurance Policy Holder. This process is based on the intensive data exchange between both parties. Character of these data can be very diverse and usually includes structured information transmitted via web forms or audio telephone communications, as well as unstructured or weakly structured data delivered in form of free texts, repair invoices, service offers and photos of damaged objects. The latter ones, namely photos, are particularly valuable because they appear at early stages of the settlement process. Photos are information rich and can be taken by Policy Holders without any special equipment or skillset needed. Therefore, teaching computers to understand images** and to automatically extract damage/repair relevant information is a crucial step towards enabling high-quality automation of the claim settlement process.

An ideal photo analysis system (Figure 1) would accept an image of a presumably damaged property object as input and in response returns (i) a list of insurance relevant objects present in the image, (ii) state of each objects (intact, damaged, old etc.) and (iii) costs associated with the repair or replacement of damaged objects. Addressing these questions automatically, by a computer system, requires unprecedented level of integration of expert knowledge into artificially intelligent (AI) system and would offer significant advantages to all stakeholders of the claim settlement process by making it faster, more accurate and more transparent.

* andrey.lutich@sachcontrol.de

** In the context of claim settlement, words “images” and “photos” are used interchangeably because radar, LIDAR etc. imaging techniques are not in common use. We stick to this convention in this white paper.

Figure 1.: Sketch of an ideal image understanding process



Recent advances in image understanding by means of deep convolutional neural networks, at first glance, seem to offer all necessary tools and methods to construct and train such AI black boxes. However, in practice, training such a system end-to-end is not feasible because of the extreme complexity of the "conclusion making process" while mapping images and the sets of desired answers. To make a computer system to learn this complexity from scratch using any state-of-the-art deep learning technique (e.g. backpropagation, reinforcement learning, adversarial learning etc.) would require enormous amount of training data samples. These data currently don't exist and is too expensive to generate. Furthermore, the black-box approach typically lacks the reasoning component on the predictions made, which is a significant drawback in the context of the negotiation character of the settlement process. Although, in many cases bare accuracy performance of AI systems may surpass human experts, the prediction of such systems is hard to use for negotiations because of the missing argumentation that humans are used to.

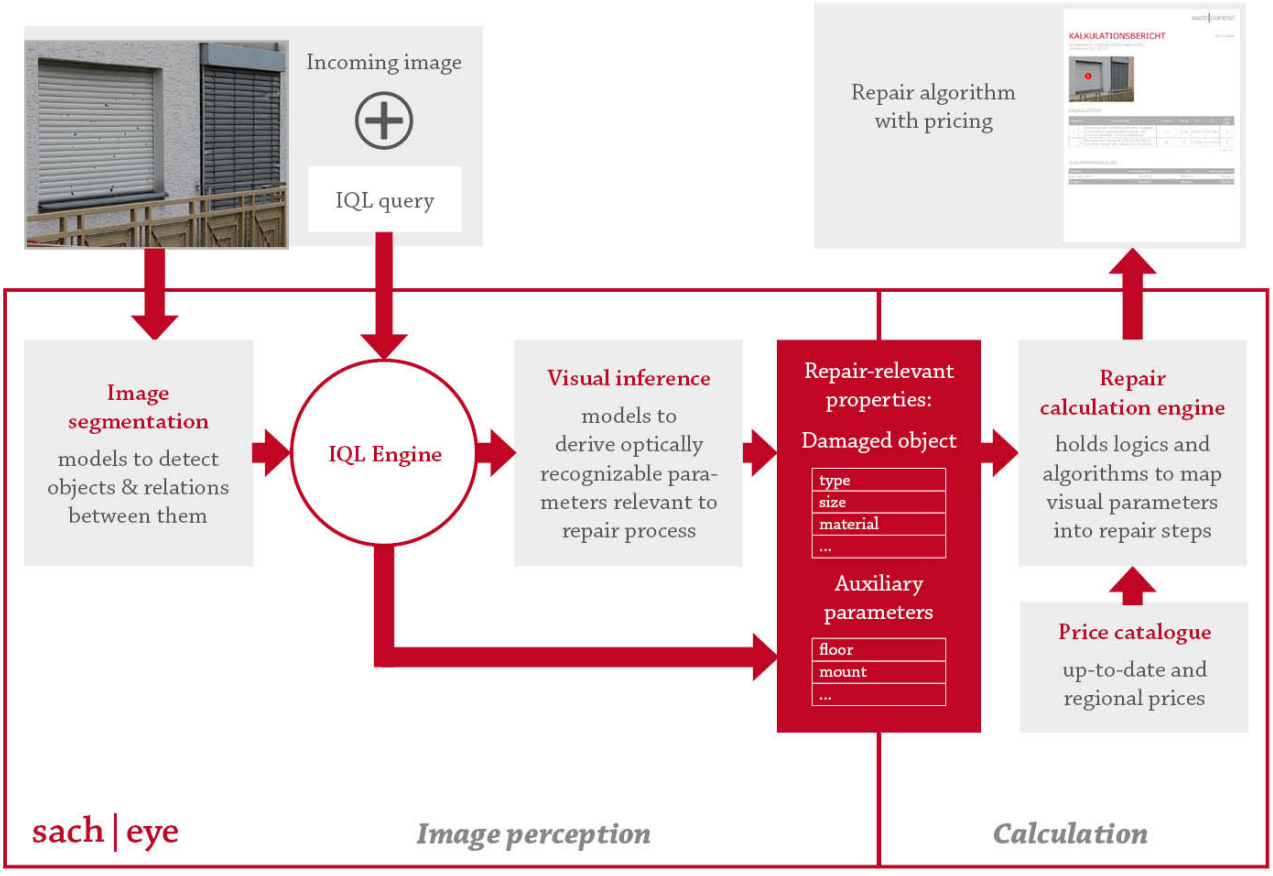
To overcome these obstacles, we have designed and trained modular image interpretation system which is constructed to allow separate training of individual models and is easily expandable. On the following pages we review major components of our image understanding method, current possibilities and challenges.

sachleye – image understanding AI system

To reduce the amount of data required to setup image-understanding/repair-pricing algorithms and enable interpretability of the results we have designed our system (sach|eye) in a modular fashion (Figure 2). At first, we split the task of deriving pricing from images in two completely isolated parts by introducing a formal vector of repair-relevant properties. This vector serves the purpose of decoupling image perception part from the repair calculation algorithms. The image perception part

is responsible for image understanding by means of image segmentation models. It also does extraction of qualitative and quantitative properties required for damage understanding and repair estimation. The repair understanding part accepts as input structured data extracted by the perception and fed through the interface defined by the vector of repair-relevant properties. This part of the system is responsible for deriving repair algorithms based on the object properties and repair-price catalog. It is based on the rules and repair logic defined by human experts in the domain of property construction and repair. The input required by this part defines elements and structure of the interface between both parts of the system. In this white paper, we focus on the perception algorithms of the sach | eye.

Figure 2.: sach | eye – image understanding and repair cost estimation workflow

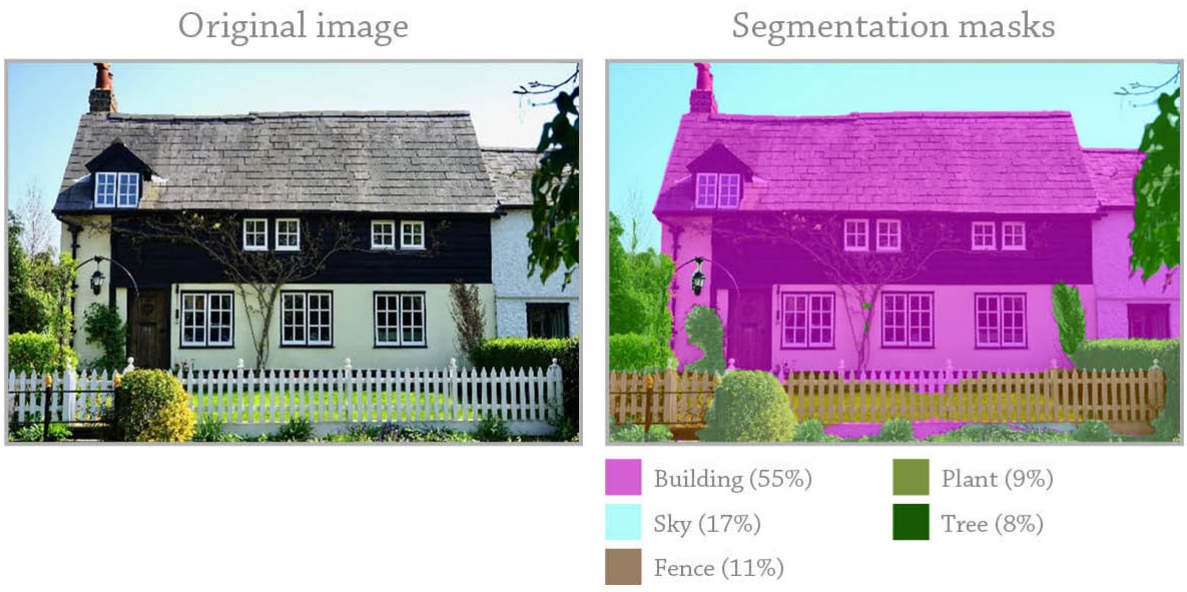


Perception part of the sach|eye includes two types of models. First type models are based on images segmentation methods and primarily make the job of identifying and isolating objects of interest in images. These models are architectured to function in a layered/hierarchical way. The hierarchy of models is essential for being able to extract relationships between objects and their parts. It is also prerequisite of allowing fine-tuning object selection in images using Image Query Language Engine. It makes use of the spatial relations between objects on different hierarchy levels. Models of second type are applied to the objects identified by segmentation models (regions marked by these models within original image). These models are aiming to predict actual properties of detected objects. These models are of regressor and/or classifier type: making predictions of type, size, material etc.

Image segmentation

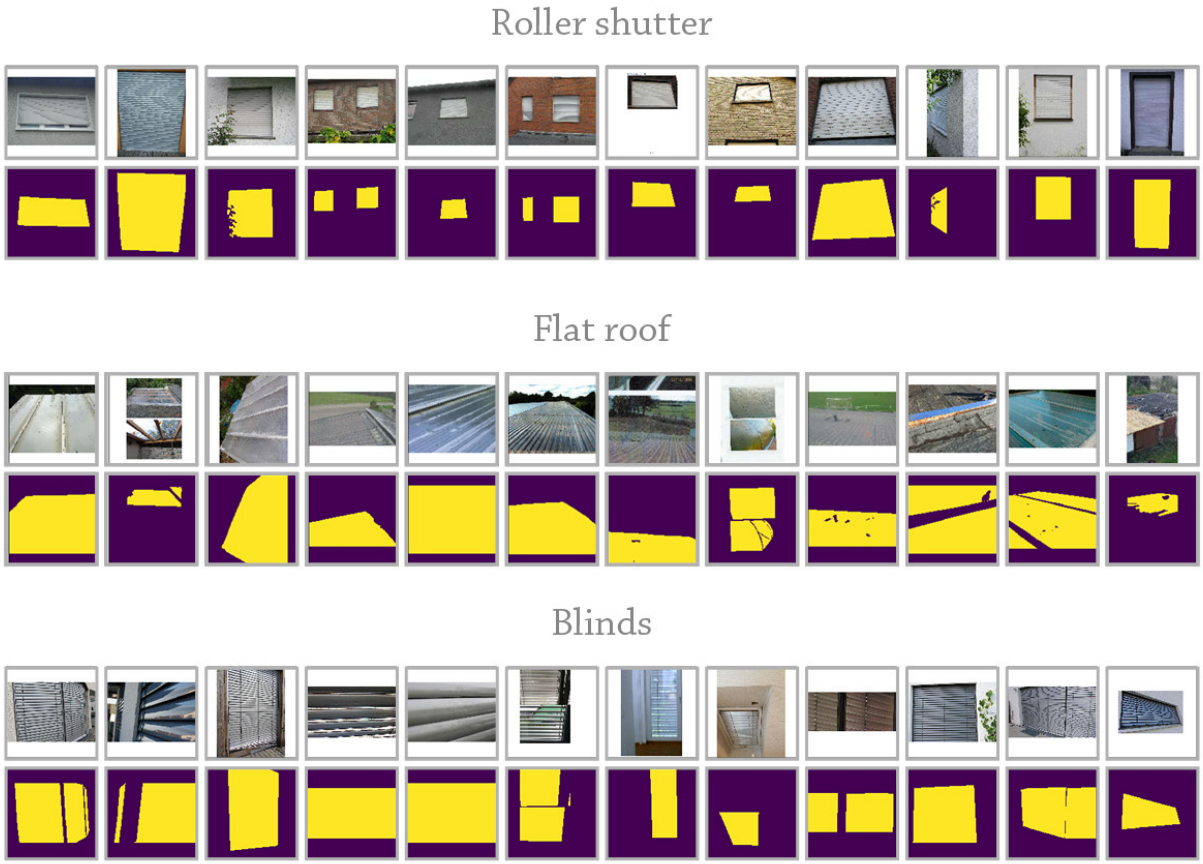
Image segmentation is the process of image classification at the pixel level. As a result, every pixel of an incoming image is assigned a label out of those the model is built for (Figure 3). We have evaluated several award-winning image segmentation architectures suitable for semantic scene parsing (ade20k challenge leaderboard [1]) and have selected DeepLab [2] as a baseline model for further fine-tuning and modifications. In contrast to image classification, where labels are assigned to entire images, image segmentation results provide considerably deeper insight into image content. For example, having segmentation masks at hand makes possible to estimate main object being photographed by comparing relative area occupied by various objects in the image. Objects occupying larger areas in an image are likely to be playing important role in the scene being photographed. Objects taking just few percent on the image are often, but not always, playing minor role. Clearly, having access to relative areas occupied by objects in images enables much more granular business logic compared to entire image classification.

Figure 3.: Image segmentation example



In general, training any image segmentation model, regardless of its internal structure, requires massive pixels-level annotated data. There have been several image datasets annotated and released under Creative-Commons-License by the computer vision community at different years, in different quality and based on every different annotation convention. These data after thorough cleaning, pre-processing and unification is a basis to teach our system to understand general-purpose objects like building, person, vehicle etc. Although preprocessing available data to a common standard is hard, generating sufficient amount of training data for new, less common, object types represents much bigger challenge. This challenge is two-fold. At first, one needs enough images of an object of interest with large degree of variation in the object itself, view angle, illumination and camera properties. Possessing the images might not necessarily be a challenge for businesses having to deal with photos in context of their currently existing property-related repair processes but selecting the right ones out of the entire data is trickier. Second, selected images must be consistently annotated across newly added and previously existed objects. High quality, dense and consistent image annotations are the key to train well-performing segmentation models. Several randomly selected examples of damaged property objects (roller shutters, flat roofs and blinds) along with their annotation masks from our dataset are shown in Figure 4.

Figure 4.: Random examples of damaged objects with annotations



To large extent, our image database consists of real-world photos of damaged property objects in their natural context. These images are very diverse, have strongly varying quality and camera orientation settings, as well as light conditions.

Images understanding

Image segmentation models are used to assign semantic labels on the pixel level. This is a very granular and insightful approach, but still it is rather simplistic compared to the depth of image understanding required to drive business applications. Stating objects present in images is sometimes helpful, but in most business relevant cases is insufficient. Often, actual understanding of scenes depicted by images means understanding of relations between objects in images. Mainstream approach to this problem is to map images to image captions. In other words, training hybrid CNN-RNN neural networks for captioning images by free-form text sentences as a human would do. These smart algorithms with impressive performance can almost perfectly learn to mimic humans describing image content (e.g. [3]). In fact, these methods map unstructured information from images into again unstructured, free-form text captions. These captions are hard be used as input to downstream business processes and, therefore, first must be converted to a more structured form. Our approach is different; we avoid the step of generating text captions from images but extract structured information from images directly. Image segmentation models is the basis and the very first step of this extraction.

State-of-the-art image segmentation model architectures designed to assign individual image pixels to single labels. Effectively, every pixel and every object (group of connected pixels with the same label) segmented in the image may belong to one and only one label at a time. However, in contrast to this simplified treatment, our, human perception and interpretation of images relies strongly on "seeing" some objects belonging to different semantic categories at the same time, or "understanding" one being a part of another one.

To artificially implement this important peculiarity of human image perception process we train several segmentation models that enable assignment of the same pixel or group of pixels in an image to different labels. Our current implementation includes three layers of segmentation models. Arbitrarily, they can be split to three levels according to their main function: (L0) general scene understanding, (L1) object-of-interest detection and (L2) damage localization (Figure 5).

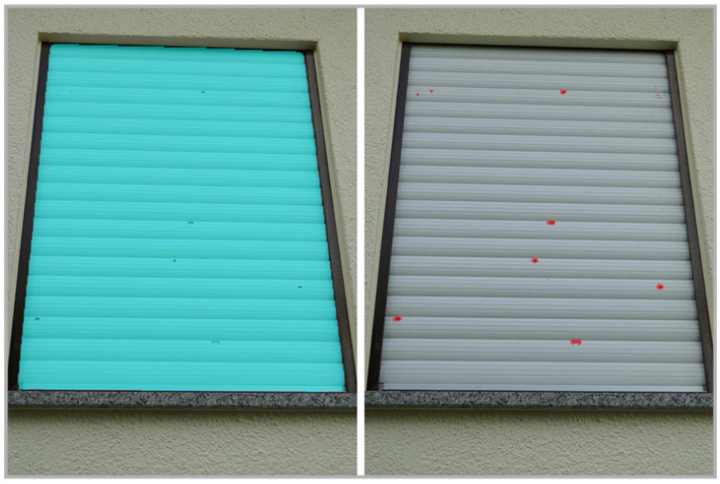
Figure 5.: Image segmentation at different granularity levels

To combine and efficiently use segmentation results generated at different levels we have developed formalized approach of converting arbitrary questions in normal human language to images into queries. We named it Image Query Language (IQL).

Image Query Language

The idea of the Image Query Language arose from the necessity to find spatial relations between various segmented objects in images for addressing questions posed to images. IQL's core is based on Boolean operations (Or, And, Not, Xor) applicable to objects discovered by the segmentation models on a per object basis. IQL also includes other mathematical helper functions (Area, Count etc.) as well as spatial relation functions (Interact, Near, NotNear etc) that can be applied as additional constrains to 2D geometrical shapes. Figure 6 illustrates an example of the same question to an image in English language and using IQL formalism.

Figure 6.: Posing a question to image using IQL



Is shutter damaged?

English language: Yes, shutter has holes over its' surface

Image Query Language: True, **area**(shutter U holes) != 0

IQL engine operates over default output of the image segmentation models and can optionally be invoked in the moment when an image is passed to sach|eye. Effectively IQL is a language that an external user can use to program sach|eye to return specific objects discovered in an image according to specified criteria. This approach offers unprecedented flexibility to users by allowing almost arbitrary combinations of predefined atomic objects known by segmentation models to be used for fine-selection process. In fact, IQL decouples business logic applied to the results of image segmentation from the process of image segmentation itself.

Visual Inference

Once objects of interest have been cut out by segmentation models and fine-selected using IQL engine they are sent to the Visual Inference Module where essential object parameters are inferred. These parameters are defined by domain experts based on the knowledge needed to perform repair calculation. Depending on a particular object, these parameters may include material, surface type, method of mounting, color, object dimensions etc. Training models to predict qualitative parameters like material, profile shape etc. requires moderate amount annotated data (1-10k training examples) to achieve superhuman accuracies in classification setting. This amount of data is feasible to acquire in most of the real-world cases. With this amount of annotated data, our model to classify roller shutter material between plastic and aluminum reaches 84.6% accuracy on the test dataset.

Building models to predict actual absolute dimensions of arbitrarily photographed objects is a challenge. Indeed, the process of taking a photo compresses information of the 3D world down to two dimensions of a planar image. Most importantly, depth information (distance to objects being photographed) is not recorded by standard photo cameras, which makes impossible to map distances in the image plane to real-world dimensions. Assuming the depth information is not available, the only available option to enable this mapping is to use some references present in an image itself. We use deep convolutions neural networks in the regression setting to learn these references using expert annotated object dimensions data on the per-object basis. This approach works surprisingly well for objects with clear, characteristic and optically detectable signatures of object's lateral dimensions. This method has optimal performance on frontal view images (take from angles close to perpendicular to object's surface). Oblique view Images complicate correct estimation for absolute lateral dimensions of objects in images for both human experts and computer vision systems. Figure 7 illustrates that although in original photo (a) image window 1 looks of similar size (if not larger) as window 2, in reality window 2 is 50% wider, which becomes obvious only after performing perspective correction (b).

Figure 7.: Perspective has strong influence on absolute dimensions perception

Perspective correction in the example in Figure 7 is a trivial task when there is a 2-dimensional reference object placed in the same plane as the objects of interest (windows surface in this case). In the above example A4 page is used to perform perspective correction and to map corrected image pixel dimensions to real-world scale. Using A4 page as reference allows measuring surface areas of planar objects in images with relative error better than 10% at typical conditions of photographing damaged windows and facades.

Strictly speaking, if there is no predefined reference object in an image, it is not possible to read out absolute object dimensions because of irreversible loss of information. On another hand, domain experts are able to make reasonable estimates of object sizes and areas based on indirect references available in images as well as their knowledge of dimensional constrains and possible size options. In attempt to mimic this estimation process we train CNN-based regression models that accept an image of a particular pre-defined object as input and infer lateral dimension and/or area of the object's surface. For the roller shutters the best area prediction model we built has relative error of 24.5%. It is calculated as average of relative absolute deviation of annotated and predicted rol-

ler shutter area over the entire test set. After multiple attempts to improve model performance we discovered that the accuracy is data limited. In other words, human experts are not always able to annotate object dimensions in images reliably and reproducibly. To confirm this hypothesis, we have asked three domain experts to annotate areas of the same roller shutters in our test set. We found average relative expert-to-expert error to be 33.5%, which means that model outperforms expert opinion dispersion by almost 10%. By analyzing annotations done by different experts, we discovered that expert opinion dispersion correlates with photographing conditions. Frontal view images have lower opinion dispersion, whereas, images taken at oblique angles and having pronounced perspective result in stronger variation of expert's judgements.

Summary

sach | eye is the image understanding AI system based on a combination of computer vision and machine learning algorithms as well as expert knowledge in the domain of property insurance and property repair. sach | eye is a modular system, where image perception engine and automated damage repair calculator are fully decoupled by means of a vector of repair-relevant parameters. Image Query Language enables external users to program the image perception core for fine-tuning desired output. Currently, visual inference and automated repair estimation is supported for images of roller shutters, blinds and flat roofs. Both human experts and sach | eye have considerably lower accuracy of object lateral size estimate on images taken at oblique angles. Placing simple, pre-defined reference objects in damage photos improves accuracy of this estimate by 3-fold for roller shutter case.

References:

1. http://sceneparsing.csail.mit.edu
2. https://arxiv.org/abs/1606.00915
3. https://arxiv.org/abs/1412.2306