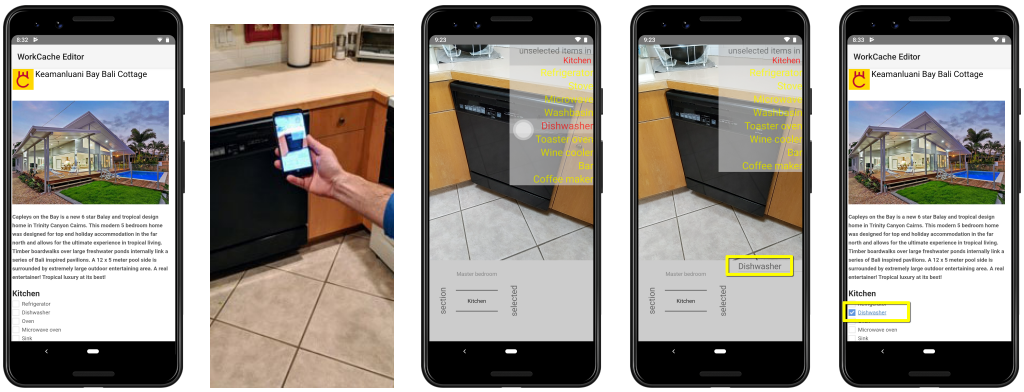
Documenting Physical Objects with Live Video and Object Detection

Scott Carter, Laurent Denoue, Daniel Avrahami
carter,denoue,avrahami@fxpal.com
FX Palo Alto Laboratory, Inc.
Palo Alto, CA



(a) (b) (c) (d) (e)

Figure 1: (a) A template document requires the end user to document different items in a rental listing. When the user first opens the mobile app, the system extracts all of the requested items for the entire document. (b) The user walks to the kitchen and selects "kitchen" in the section filter list to see only those items in the app. (c) The app detects the dishwasher in view and highlights its label in red. (d) The user selects this label and the system automatically saves a high quality image and moves the item label to the selected list. (e) The system automatically updates the linked template to indicate that the item is documented and inserts a link to the photo of the item.

ABSTRACT

Responding to requests for information from an application, a remote person, or an organization that involve documenting the presence and/or state of physical objects can lead to incomplete or inaccurate documentation. We propose a system that couples information requests with a live object recognition tool to semi-automatically catalog requested items and collect evidence of their current state.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

KEYWORDS

mobile, object recognition, template filling

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10.
https://doi.org/10.1145/3343031.3350581

ACM Reference Format:
Scott Carter, Laurent Denoue, Daniel Avrahami. 2019. Documenting Physical Objects with Live Video and Object Detection. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3343031.3350581

1 INTRODUCTION

There are many situations that require people to respond to requests for information from a remote person, application, or organization that involve documenting physical objects. Furthermore, in many of these cases it is important that the response also include evidence (e.g., photographs, video, or meta-data). For example:

- When listing real estate, the realtor or seller needs to document different features of the listed house or lot, such as appliances, the relationship with nearby property, the condition of fixtures and other materials, and so on.
- People preparing properties for short-term rentals have similar concerns, but may also want to collect evidence of the presence and/or condition of many items (both before and after the rental period) for insurance purposes.
- Insurance organizations often require claimants to collect photographic and other evidence before filing (e.g., photos of automobile damage).

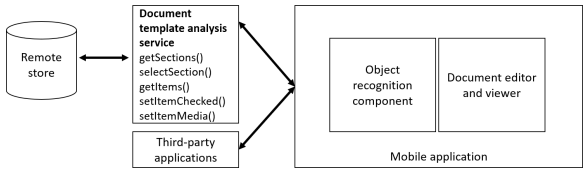


Figure 2: The mobile application can get item and media information from and send updates to third-party services or a document template analysis service running remotely.

- People selling objects may need to document certain aspects of their item before they are allowed to publish their listing on public web sites.
- Inspectors and field service technicians may need to document particular issues before they can file a work order or verify that work is complete.
- In surgical settings, ensuring that all surgical instruments are collected and accounted for after an operation can be critical for avoiding surgical adverse events (SAEs).
- Doctors may request that patients document issues such as wounds, skin disorders, limb flexibility, or other medical conditions before meeting with patients (especially when meeting patients remotely via telemedicine interfaces).

Typically, people completing these requests operate off of a static list that can often lead to incomplete or inaccurate documentation. In this work, we describe a system that couples information requests with a live object recognition tool to semi-automatically catalog items and collect evidence of their existence and current state.

2 SYSTEM

Our system allows end users to scan an environment to catalog and capture media of objects-of-interest. The system combines three components: acquiring the information request, detecting objects with live video in a mobile application, and responding to the information request (Figure 2).

2.1 Information request acquisition

External systems can send requests to a mobile application with a JSON payload that includes text descriptions of the required items. The payload can optionally include extra information such as whether each item is currently selected, the type of the item (such as a simple radio box item or a photo), as well as a description of a group or section to which a item belongs. The system can also generate information requests from document templates. We built a document analysis tool to parse Markdown-based documents to extract certain items, such as radio-boxes, as items.

2.2 Detecting objects with live video

Our mobile application ingests information requests to populate a video-based user interface. When a user selects a document template from a list, the system generates a JSON payload with item information, which it forwards to the live viewer. All of the possible items appear in the upper-right, overlayed on the live video. If the JSON payload included tokens with different sections (e.g., radio boxes from different sections of a document template), the viewer shows a selectable list of sections on the lower-left. Selecting a section filters the list of current items, allowing users to select items with similar names in different sections of the document.

As the user scans the viewer around their environment, the live viewer runs a separate thread analyzing frames with an object recognizer. The system uses the TensorFLow Lite framework with the Inception-v3 image recognition model trained on ImageNet, which includes about 1000 classes of items. As objects are recognized, they pass through two filters. First, a configurable threshold filter eliminates objects for which the system has low confidence. Objects that pass the first filter are then filtered against the items from the information request. To pass this filter, we first tokenize and stem each item and recognized object description. We then require that at least one token of each item matches at least one token from the object recognized (e.g., “Coffee filter” would match “Coffee”, “Coffee pot”, etc.). If the object passes the second filter, the application automatically caches a photo the object and highlights the item(s) to which it corresponds in the interface (Figure 1c). The application can optionally cache a short video of the object.

2.3 Responding to the information request

Once an item is highlighted, the user can click on it to select it. Once selected, the item is removed from the upper item list and moved to the selected list below (Figure 1d). On a background thread, the application also forwards the selected item description and metadata as well as the cached photo to the requesting service. In our system, the backend service ingests the selection and updates the corresponding document template on-the-fly (e.g., it selects the items corresponding radio box and injects a link to the uploaded photo, see Figure 1e).

Users can also deselect an item at any time in the mobile application. This action sends a deselection event back to the server.

Finally, our mobile application also includes a document editor and viewer so that users can confirm updates made by the object recognition component.

3 RELATED WORK

Insurance companies are integrating some recognition into their mobile apps for customers filing claims. However, this is limited to OCR (for determining, for example, the make and model of a car) [3], or guiding users to capture better quality shots of damaged areas [4–6]. Other related systems recognize road signs from videos and add them to GIS data [7].

Unlike other apps, our tool integrates live object recognition to semi-automatically catalogue items. This approach provides a stronger guarantee that an object was captured than similar nudging approaches.

4 CONCLUSIONS AND FUTURE WORK

Our video-based object recognition tool allows end users to document physical items rapidly. However, while the image recognition module in the current embodiment is useful for some cases, it may be too generic and broad for every case. We can take advantage of well-known approaches for creating image recognition models for more other targeted domains, such as retraining [2] or transfer learning [1], to extend the usefulness of the app.

REFERENCES

[1] 2018. Transfer Learning for Image Classification using Keras. https://towardsdatascience.com/transfer-learning-for-image-classification-using-keras-c47ccf09c8c8.

[2] 2019. Image retraining. https://www.tensorflow.org/hub/tutorials/image_retraining.

[3] 2019. State Farm claims app adds object recognition for simple submission. https://www.retaildive.com/ex/mobilecommercedaily/state-farm-claims-app-adds-object-recognition-for-simple-submission.

[4] 2019. Tractable. https://tractable.ai/products/car-accidents/.

[5] 2019. ViewSpection. https://www.viewspection.com/.

[6] S. Carter, J. Adcock, J. Doherty, and S. Branham. 2010. NudgeCam: Toward Targeted, Higher Quality Media Capture. In *Proceedings of the International Conference on Multimedia (MM '10)*. ACM, 615–618.

[7] S. Šegvic, Z. Brkić, K. Kalafatić, V. Stanisavljević, M. Ševrović, D. Budimir, and I. Dadić. 2010. A computer vision assisted geoinformation inventory for traffic infrastructure. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems*. IEEE, 66–73.