CamaLeon: Smart Camera for Conferencing in the Wild

Laurent Denoue, Scott Carter, Chelhwon Kim
{denoue,carter,kim}@fxpal.com
FX Palo Alto Laboratory, Inc.
Palo Alto, CA

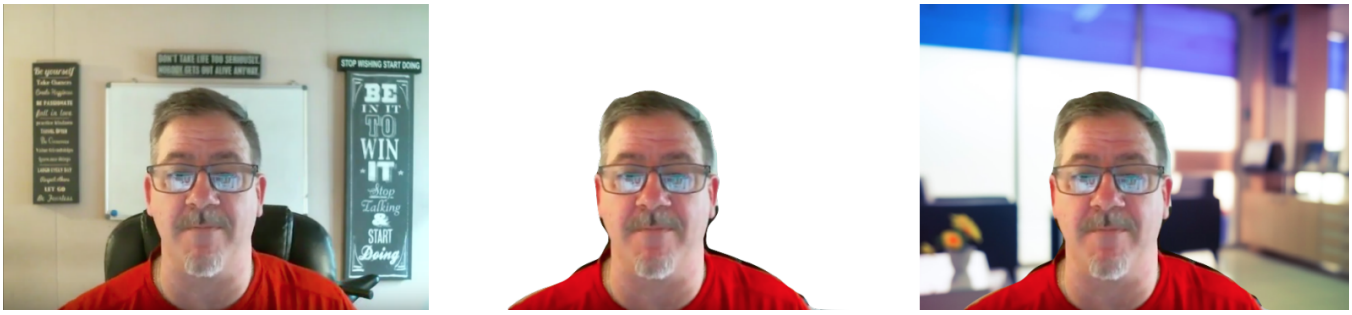

Figure 1: CamaLeon system: (left) original, (middle) background removed, and (right) remote background copied behind.

ABSTRACT

Despite work on smart spaces, nowadays a lot of knowledge work happens in the wild: at home, in coffee places, trains, buses, planes, and of course in crowded open office cubicles. Conducting web conferences in these settings creates privacy issues, and can also distract participants, leading to a perceived lack of professionalism from the remote peer(s). To solve this common problem, we implemented CamaLeon, a browser-based tool that uses real-time machine vision powered by deep learning to change the webcam stream sent by the remote peer. Specifically, CamaLeon dynamically changes the "wild" background into one that resembles that of the office workers. In order to detect the background in disparate settings, we designed and trained a fast UNet model on head and shoulder images. CamaLeon also uses a face detector to determine whether it should stream the person's face, depending on its location (or lack of presence). It uses face recognition to make sure it streams only a face that belongs to the user who connected to the meeting. We tested the system during a few real video conferencing calls at our company in which two workers are remote. Both parties felt a sense of enhanced co-presence, and the remote participants felt more professional with their background replaced.

CCS CONCEPTS

• **Information systems → Web conferencing**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

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

ACM ISBN 978-1-4503-6889-6/19/10.
https://doi.org/10.1145/3343031.3350583

KEYWORDS

video conferencing, person segmentation, face detection, real-time

ACM Reference Format:
Laurent Denoue, Scott Carter, Chelhwon Kim. 2019. CamaLeon: Smart Camera for Conferencing in the Wild. 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/10.1145/3343031.3350583

1 INTRODUCTION

Knowledge work happens in many places, such as cafes, home offices, or transportation. When these "in the wild" users need to conduct live meetings with a camera, they can appear unprofessional and not present. This is especially true when they join a meeting where most participants are co-located in a regular office meeting room.

With CamaLeon (Figure 1), we offer a completely browser-based solution that dynamically changes the "wild" background into one that resembles that of the office workers (see demonstration at [1]).

The tool offers 3 modes: 1) no background replacement, 2) a static image chosen by the remote participant and 3) the "CamaLeon" mode where the remote participant's background copies the look of the other peer's background.

In the next section, we detail the techniques developed to implement a real-time, browser based tool.

2 REAL-TIME BACKGROUND REMOVAL

2.1 Deep Learning for person masks

We treat background removal as its dual problem: identifying the person's silhouette. Background removal has a long history in computer vision, but traditional methods are not robust enough for many real-world settings [10]. For example, they assume a fixed camera, fixed illumination, none of which are realistic for our purposes, e.g., working in a cafe, low light conditions at home, working while mobile using a laptop.

However, modern deep learning networks, especially Convolutional Neural Networks (CNNs), have shown to be robust in these conditions. We thus designed a lightweight semantic segmentation network based on the UNet architecture [5]. Specifically, we adopted a small C8-C16-C32-C64-C128 encoder, followed by the corresponding decoder CD64-CD32-CD16-CD8[1], trained with the Adam optimizer and a binary cross-entropy loss.

We trained the network on 1672 images from the Flickr dataset [6]. This dataset appropriately contains images of people's face and shoulder, along with their ground-truth masks (black pixels for background, white for person). We trained the UNet in Keras, and converted it to TensorFlow.js [7] in order to perform inference in the browser.

After several iterations on the network, we settled on a net that has only 485,817 parameters, and weights only 2MB once converted to TensorFlow.js. On a Mac Book Pro, the inference runs at over 20 frames per second, enough to enhance our WebRTC, web-based conferencing system [2].

2.2 Face Detection

Since the training data only contains head and shoulders, the system would fail when the person is sitting too far away or off-center from the camera view. To accommodate for this problem, we use a real-time face detector [3] that returns the bounding box of the person's head. Using it, we send a cropped region to the UNet for background removal.

Every so often, we also run the face recognition model to enhance privacy and confidentiality of the meeting, making sure that the user detected in the webcam is the authenticated user. For example, if the user moves away from the webcam view, revealing another person's face in a cafe, the remote participants will not see their face. To enhance speed, we precompute and locally store facial keypoint features of our test users from our laboratory.

2.3 Implementation Details

Modern web browsers contain a WebRTC library, allowing to conduct peer-to-peer web meetings. In order to modify the captured stream, we first draw incoming webcam frames into a CANVAS object. The CANVAS's stream is then plugged into the WebRTC stream as a source object: everything that is drawn into the CANVAS is streamed to the remote participants.

Given a new webcam frame, we first apply the face detector. The resulting box is fed into the UNet to infer background/foreground pixels. Depending on the current mode (static image or CamaLeon mode), the foreground pixels are copied over a static image or the remote background captured by the remote participants.

2.4 CamaLeon background

In order to overlay the person's body (face+shoulder) over the remote background, the system accumulates a model of this background over time. Specifically, every two seconds (chosen based on current hardware speed to keep the system running above 20 FPS), the remote peer sends the background image to the local user. In this way, the recipient does not need to run UNet twice. This image is blended into the current model of the background image using a running average.

Initially, this background model contains large holes, corresponding to where remote users show up in their webcam. To solve this issue, we tried image inpainting techniques [8, 9], but none was fast enough for our purposes. Also, filling out a large area in the center of the frame (where the user typically shows) is not a typical use-case for image inpainting.

We solve this problem by translating and scaling the local user's position to cover the pixels occupied by the foreground mask of the remote peer. The background model gets updated as remote peers move, allowing the system to position the local user at their actual position.

A nice side effect of this feature is that both peers' faces initially occupy a similar area on the screen, which might further enhance their sense of being together (immersion).

3 RESULTS AND FUTURE WORK

We tested the prototype with 3 real-meetings. First we tested with a remote worker who connects from his home office in Europe, 9 hours away from our US-based laboratory. Anecdotal evidence shows that he felt more secure in how he presented himself. The US-based peer also liked the system, preferring the CamaLeon mode to the static image.

We also tested with two co-workers who were working from home, one located in Europe and the other in the US. In that case, the CamaLeon mode was not as valuable: both parties knew each other well and could understand their respective settings.

In the third case, we tested the system with a remote participant connecting to a board meeting where around 20 co-located users could see him. In that case, the CamaLeon mode made the remote worker feel much more secure in his professionalism than have either a static or no background replaced.

The system runs at 20 FPS on a Mac Book pro, but it still has issues with fine masks. Blending the contours of the body pixels could help improve the perception of immersion.

Also, we noticed that people wave their hands during meetings, e.g., to explain a concept or point at materials displayed on a side display. The current UNet model we trained does not detect arms and hands correctly. We are now replacing it with BodyPix [4], a deep network that also runs in real-time in modern browsers, but has been trained on full body masks.

Real-time image inpainting would give more freedom to the placement of local users: currently, the system locates them to overlap the foreground pixels of the remote participants, which is not always ideal. Another option is to use a patch of the remote background, stretch it across the local peer and apply super resolution. Although no user complained about the CamaLeon mode copying their background, a larger study might reveal some discomfort by some users when they see their office background twice.

Finally, we would like to deploy our web-based tool in real settings, for example an online interviewing platform (such as CoderPad.io) and test several conditions: no background, static image, blurred background (as offered in recent web meeting tools such as Microsoft Teams), and our CamaLeon mode.

[1] Ck denotes Convolution-ReLU-Convolution-ReLU-Downsampling layer with k filters, and CDk denotes Convolution-ReLU-Convolution-ReLU-Upsampling layer.

REFERENCES

[1] Laurent Denoue, Scott Carter, and Chelhwon Kim. 2019. CamaLeon Online Demonstration. https://docuchat.fxpal.com/camaleon/callpeer.html

[2] Andreas Girgensohn, Jennifer Marlow, Frank Shipman, and Lynn Wilcox. 2015. HyperMeeting: Supporting asynchronous meetings with hypervideo. In *Proceedings of the International Conference on Multimedia*. ACM, 611–620.

[3] Vincent Mühler. 2019. Face-API.js. https://github.com/justadudewhohacks/face-api.js.git

[4] Dan Oved and Tyler Zhu. 2019. BodyPix. https://aijs.rocks/inspire/bodypix/

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 234–241.

[6] Xiaoyong Shen, Aaron Hertzmann, Jiaya Jia, Sylvain Paris, Brian Price, Eli Shechtman, and Ian Sachs. 2016. Automatic portrait segmentation for image stylization. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 93–102.

[7] Daniel Smilkov, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang, Shanqing Cai, Eric Nielsen, David Soergel, Stan Bileschi, Michael Terry, Charles Nicholson, Sandeep N. Gupta, Sarah Sirajuddin, D. Sculley, Rajat Monga, Greg Corrado, Fernanda B. Viégas, and Martin Wattenberg. 2019. TensorFlow.js: Machine Learning for the Web and Beyond. arXiv:cs.LG/1901.05350

[8] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Free-Form Image Inpainting with Gated Convolution. *arXiv preprint arXiv:1806.03589* (2018).

[9] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Generative Image Inpainting with Contextual Attention. *arXiv preprint arXiv:1801.07892* (2018).

[10] Cha Zhang, Li-wei He, and Yong Rui. 2010. Background blurring for video conferencing. US Patent 7,783,075.