CORE-Lens: Simultaneous Communication and Object REcognition with Disentangled-GAN Cameras

Ziwei Liu[1*] Tianyue Zheng[2*] Chao Hu[1] Yanbing Yang[1,3] Yimao Sun[1,3] Yi Zhang[1,4]
Zhe Chen[5] Liangyin Chen[1,3] Jun Luo[2]
[1]College of Computer Science, Sichuan University, China
[2]School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore
[3]Institute for Industrial Internet Research, Sichuan University, China
[4]School of Cyber Science and Engineering, Sichuan University, China
[5]AIWiSe, China-Singapore International Joint Research Institute, China
Email: liuziwei0901@stu.scu.edu.cn, {tianyue002, junluo}@ntu.edu.sg, yangyanbing@scu.edu.cn

ABSTRACT

Optical camera communication (OCC) enabled by LED and embedded cameras has attracted extensive attention, thanks to its rich spectrum availability and ready deployability. However, the close interactions between OCC and the indoor spaces have created two major challenges. On one hand, the stripe pattern incurred by OCC may greatly damage the accuracy of image-based object recognition. On the other hand, the patterns inherent to indoor spaces can significantly degrade the decoding performance of *reflected* OCC. To this end, we propose CORE-Lens as a *pipeline* to make the mutual interference transparent to existing OR and OCC algorithms. Essentially, CORE-Lens treats the two challenges as two sides of a signal mixture issue: the signals transmitted by OCC get mixed with background images so well that their features become entangled. Consequently, CORE-Lens exploits the idea of *disentangled representation learning* to separate the mixed signals in the feature space: while the GAN-reconstructed clean background images are used to perform object recognition, OCC decoding is conducted on the residual of the original image after subtracting the reconstructed background. Our extensive experiments on evaluating the real-life performance of CORE-Lens evidently demonstrate its superiority over conventional approaches.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Computing methodologies → Computer vision**; • **Networks →** *Wireless access networks.*

KEYWORDS

Optical camera communication (OCC), VLC, ISAC, object recognition, disentangled representation learning, GAN.

ACM Reference Format:
Z. Liu, T. Zheng, C. Hu, Y. Yang, Y. Sun, Y. Zhang, Z. Chen, L, Chen, and J. Luo. 2022. CORE-Lens: Simultaneous Communication and Object REcognition with Disentangled-GAN Cameras. In *The 28th Annual International Conference On Mobile Computing And Networking (ACM MobiCom'22), October 17–21, 2022, Sydney, NSW, Australia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3495243.3560526

1 INTRODUCTION

As an important branch of *visible light communication* (VLC) utilizing visible light to realize ubiquitous device connection [51, 55, 57, 61, 64, 65], *optical camera communication* (OCC) has been extensively developed, piggybacking on LED lighting infrastructures as transmitters and embedded cameras (e.g., smartphone cameras) as receivers [1, 11, 13, 31, 34, 37, 63, 65]. While a camera can receive coded information transmitted directly from LED lights [11, 18, 31, 33, 40], practical applications often prefer receptions from *reflected* light transmissions [1, 13, 34, 37, 65]. However, since OCC piggybacks on lighting infrastructures, the LEDs have to perform both lighting and transmission, which in turn confines an embedded camera to act only as a receiver for OCC transmissions, since its basic functions in capturing images and performing tasks entailed by *computer vision* (CV) can be largely undermined by information coding. In fact, this observation has already been exploited to protect human users against unauthorized photographing [68].

Figure 1 describes a typical scenario of using reflected OCC for indoor VLC, where a smartphone camera also assumes the duty of

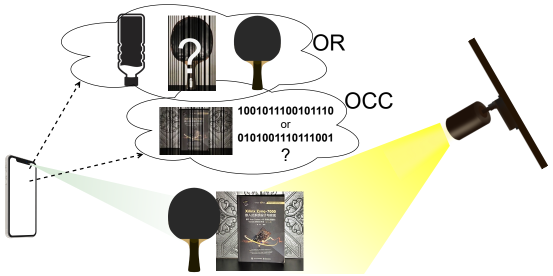


Figure 1: OCC demands an LED luminaire to emit information coded light. However, the mutual interference between the resulted light stripes and background objects affects both the object recognition (OR) accuracy and OCC decoding performance.

[*] Both authors contributed equally to this research.

ACM MobiCom'22, October 17–21, 2022, Sydney, NSW, Australia

ACM ISBN 978-1-4503-9181-8/22/10.
https://doi.org/10.1145/3495243.3560526

performing object recognition. This seemingly toy-like illustration actually represents a wide spectrum of applications including the following prominent examples. The first application is for museum or shopping mall scenarios: when users point their smartphone cameras to an art piece or a displayed item, CORE-Lens can help realizing CV functions (e.g., augmented reality for putting a hat onto user's head) while reading descriptive information delivered via OCC. Another application is efficient OCC for robots (communication and control) in public spaces (e.g., airport) or industrial facilities where a lot of background interferences may present; in the meantime, the vision of robots needs to operate properly without being affected by OCC. Last but not least, CORE-Lens may help improving the robustness of security functions leveraging CV (e.g., user identification or face recognition) under artificial interference (e.g., OCC or similarly created patterns), especially for operations carried out by mobile devices.

As these applications all demand the LED luminaire to both light a room and transmit information, the emitted light contains codes necessary to convey information. Although these codes are designed to be indiscernible by human eyes (so called flicker-free OCC), embedded cameras, with their inherent rolling-shutter effect [13], are inevitably interfered with by such information coding, if they assume duties other than OCC receivers. In particular, considering the duty of face recognition, the face in a captured image can be severely "masked" by the light stripes caused by information coding. One can readily imagine that the accuracy of recognizing both human subjects and non-human objects can be severely degraded.[1] Actually, this interference is mutual, as objects (e.g., wall papers or banded bags) with patterns may also raise the errors in OCC decoding [7, 11, 21, 34, 65]. Therefore, the major challenge behind these real-life applications is the *mutual interference* between two aspects, namely information coded light and background objects; it can damage both the basic functions of embedded cameras (especially those for CV) and the OCC decoding performance. Though separating CV and OCC in a time-divided manner [62] while leveraging ad-hoc filtering tricks [11, 21, 26, 32] to imperfectly handle OCC decoding is feasible, this straightforward solution is both inefficient and ineffective according our later evaluations.

Therefore, we aim to tackle both aspects of the challenge simultaneously in this paper, by focusing on the fundamental entanglement between the two co-located and co-existing aspects. In fact, the entanglement is the result of signal mixing caused by overlapping, and it is mandatory to separate the mixed signals so as to avoid their mutual interference. Inspired by the recent development of *disentangled representation learning* (DRL) [20, 29], we believe that the separation should be conducted in *feature space*. Compared with ad hoc imaging processing techniques directly working on image pixels, the benefit of separating signals in the feature space is twofold: i) it should be more efficient and effective, because the feature representation is much sparser yet contains critical semantics not obvious in the pixel space, and ii) while the outcome of separation is directly applicable to drive CV functions, the process could be made transparent to both CV and OCC if the background can be reconstructed based on the separation outcome. Essentially, we intend to build an *end-to-end deep learning pipeline* so that its outcome can be simultaneously used to support both CV and OCC functions, regardless of what specific types of CV (trained) model and OCC (decoding) algorithms are adopted.

To this end, we propose CORE-Lens to realize such a pipeline in resource constrained smartphones. As the first step towards an *integrated sensing and communication* (ISAC) framework [12] for VLC on smart devices, we confine the concerned CV functions to only *object recognition* (OR). CORE-Lens involves a DRL network trained via both variational inference [3] and adversarial learning [17], so as to separate the features for background sceneries from those of the coded light. Consequently, the output of this DRL network drives two functions: i) a trained OR classifier directly makes use of the background sceneries reconstructed via conditional GAN [24] for classification, and ii) the residual of the original image subtracting the GAN-reconstructed background are fed to a typical OCC decoder for information retrieval. In summary, we make the following major contributions in this paper:

- We, for the first time, identify the need for reconciling the conflicting aspects of visible light applications, namely sensing and communication, from an ISAC perspective.
- We innovate in proposing a DRL method to handle this reconciliation in feature space, in order to have an end-to-end deep processing pipeline masking the underlying details from conventional sensing and communication functions.
- We combine variational inference with adversarial learning to separate the features for background sceneries from those of the coded light, so that a trained OR classifier can directly work on the outcome.
- We further filter the residuals of an original image subtracting its GAN-reconstructed background and fed them to an OCC decoder, so that any conventional decoders can be reused with no need for ad hoc modifications.
- We conduct extensive evaluations on CORE-Lens; the results evidently demonstrate its superiority over conventional approaches for both visible light sensing (i.e., CV) and communications.
- The datasets (to be shared) resulted from our experiments, being the first of its kind, may substantially advance the research on realizing ISAC-ready VLC.

The rest of the paper is organized as follows. Section 2 explains more on the background and uses simple experiments to motivate our later design. Sections 3 and 4 respectively present the design and implementation of CORE-Lens. Extensive evaluations are reported in Section 5, along with further discussions on the advantages over existing solutions and technical limitations of our current implementations. Finally, Section 6 concludes the whole paper.

2 BACKGROUND AND MOTIVATIONS

We set up the background and motivate CORE-Lens design in this section. We first provide the technical background for simultaneous OCC and OR. Then we briefly justify the adverse effects of the mutual interference between OCC-coded light and background objects to be recognized. Lastly, we explain the basic ideas of DRL followed by a brief study on its feasibility to our solution.

[1] To avoid potential ethical concerns, we only consider non-human object recognition tasks in this paper, but the solution techniques and results can be readily extended to human identification and face recognition.

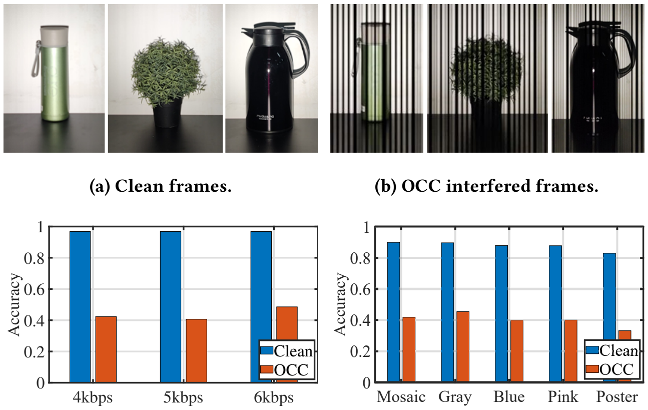
2.1 Primers for OCC and OR

OCC exploits the rolling shutter of CMOS cameras to capture information coded in light emitted from LED luminaires [13]. To be specific, since the rolling shutter exposes the frame in a column-wise manner, a CMOS camera records temporally modulated information as bright-dark stripes in the received frames [11, 18, 31]. Whereas earlier OCC systems leverage direct LED-camera link for communications (i.e., *direct OCC*) [11, 18, 31], practical application scenarios often demand an indirect LED-reflector-camera link for OCC (i.e., *reflected OCC*), in order to improve throughput while enhancing user experience [1, 34, 37, 65]. Moreover, adopting reflected OCC also enables us to perform both OCC and CV functions simultaneously with the same camera. Nonetheless, such a convenient implementation causes mutual interference between the two functionalities: while OCC decoding is known to be confused by ambient settings such as objects with patterns [7, 11, 21, 34, 65], the stripes caused by OCC also affect the CV functions seriously. Note that the seminal proposals on backscatter VLC [55, 57, 60] may become relevant upon future adoption for indoor scenarios.

As one of the major CV functions often supported by mobile apps, OR has become increasingly ubiquitous under the support of mobile devices equipped with embedded (CMOS) cameras. Though being heavily studied in CV for a few decades, OR is now taking CNN (convolutional neural network) [19, 22, 25, 28, 49, 50, 59] as the de-facto standard module, especially for mobile devices with limited resources. Essentially, a CNN slides its convolutional kernels across the images to capture information and generate summarized feature maps. As it goes deeper, more high-level features can be assembled by taking advantage of the hierarchical patterns in the input data. However, the hierarchical architecture of CNN also induces an error propagation from the input image to the classification result: if the initial steps go wrong, e.g., the network fails to locate the region of interest, or the lower-level features are strongly interfered, the final classification performance may be significantly degraded. Quite unfortunately, the OCC-coded stripes on a frame can be a strong interference source, if the frame is taken to also fulfill the function of OR. In fact, this observation has been exploited to thwart unauthorized photo shoots (and any subsequent CV processing) [68].

2.2 Confused Object Recognition (OR)

As discussed in Section 2.1, OCC transmits data by encoding them in the temporally modulated light emitted by LED luminaires, and a rolling shutter camera captures this process as striped frame images. Consequently, images captured by a camera under OCC are a mixture of background objects with foreground stripes, as shown in Figure 2b. It can be conjectured that feeding such banded images to an off-the-shelf OR algorithm (e.g., a CNN-based model) may cause significant confusion. On one hand, dark stripes may break up basic object features that would have been leveraged by OR algorithms to make correct judgements. On the other hand, the banded pattern can mislead an OR algorithm to identify an object as something non-existent in the image but sharing similar patterns (e.g., zebra or music staves) [56]. One may eliminate the foreground by integrating multiple images over time, but this inefficient solution negates our ISAC-VLC setting where OCC and OR can take place simultaneously without significant delays. Re-training an OR model can hardly be generalizable beyond the training dataset, because the interfering patterns can be highly diversified. In short, no existing proposal has effectively tackled this issue yet.



(a) Clean frames.

(b) OCC interfered frames.

(c) Different data rates.

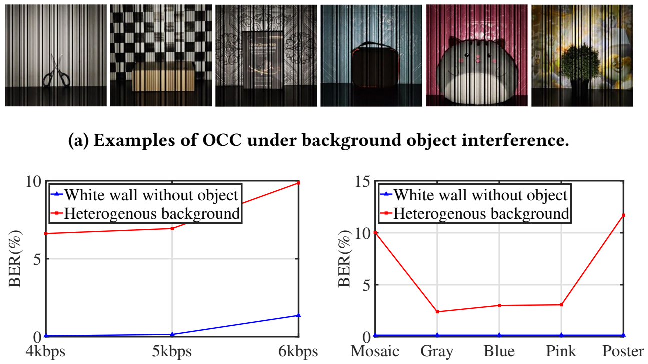
(d) Varying backgrounds.

Figure 2: OCC interference degrades OR accuracy.

We perform a set of preliminary experiments to verify the damaging effects of the OCC interference to OR. We employ GoogLeNet [50] to classify 10 classes of objects, among them are a bottle, a pot of plant, a teapot, and other daily articles. Figures 2a and 2b show these examples under normal lighting and OCC interference, respectively. We first evaluate the influence of OCC on OR under different data rates and a white background in Figure 2c, which shows the recognition accuracy dropped by more than 50% under certain data rates. In addition, the accuracy drops slightly less under a higher rate, possibly because denser stripes create lower interference. We further study the OCC interference to OR under different backgrounds in Figure 2d, when fixing the data rate at 5kbps. It can be observed that the overall accuracy under patterned backgrounds can be slightly (around 8%) less than that under a white background. In general, the accuracy is degraded by OCC interference for more than 45% regardless of the background. The consistent accuracy drop indicates that OR cannot be improved by tuning parameters or changing scenarios, hence calling for a redesign of the OR algorithm in order to remain robust against OCC interference.

2.3 Degraded OCC Performance

It is well recognized that a white and flat reflector achieves the maximum performance of reflected OCC [7, 11, 21, 34, 65]. However, such an ideal condition barely exists in practice, and real-world scenes are detrimental to OCC for many reasons. For example, the existence of 3D objects between the camera and the reflector creates additional reflection surfaces and casts shadows, causing discontinuity and distortion in the captured frame. Moreover, background heterogeneity poses another challenge: both glossy and dark materials can strongly affect reflection by drastically intensifying it in some parts while heavily attenuating it in others. In short, the twisted and altered reflection intensity caused by background heterogeneity can severely interfere with the OCC decoding performance. This damaging effect often cannot be avoided by turning



(a) Examples of OCC under background object interference.

(b) Varying data rates.

(c) Varying backgrounds.

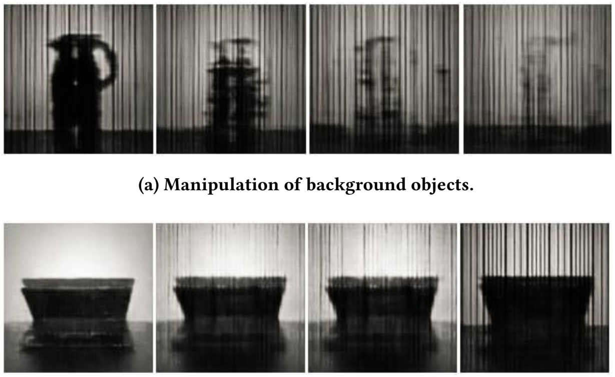
Figure 3: Background objects and patterns increase OCC decoding errors significantly.

the camera towards a "clean" area given the typical application scenarios for reflected OCC: in a department store or museum where a displayed item is lit by an OCC luminaire to convey information about it, a camera aiming to retrieve the OCC-coded information is forced to accommodate background objects. Existing

We perform another set of experiments to confirm how background objects affect OCC, with a few typical experiment setups shown in Figure 3a. In Figure 3b, we study the BER (bit error rate) degradation of Manchester decoding [13, 37, 65] caused by background interference under varying data rates. One may readily observe that the BERs are increased by approximately 8% on average under all data rates given the interference from background objects. We also study the effects of different background patterns and colors when fixing the data rate at 5kbps. The resulting BERs reported in Figure 3c show that, whereas pure color backgrounds have relatively minor impact, backgrounds with graphic patterns increase the BER by more than 10%, and the co-action of objects at different depths pushes the BERs up to 12%, rendering OCC barely operable. Therefore, making OCC more robust to various background objects is imperative.

2.4 Can Mixture be Disentangled

Ideally, in order to achieve simultaneous OR and OCC, the captured frame images should be separated into background scenes and OCC-coded patterns so that they could be processed individually. However, separating the respective components of a mixture in pixel space is an ill-posed problem for several reasons. First, unknown environmental factors (e.g., lighting conditions, background reflectivity, and background depth) make the mixing function indeterminate. Second, the randomness of the OCC-coded light makes the mixing of the two components time-varying and unpredictable. Last but not least, the existence of noise and ambient light interference complicates the separation process. Fortunately, disentangling the features of individual components can be doable because of their distinctness (e.g., textures, intensities, and shapes). Recent studies on DRL (disentangled representation learning) also show that the latent space of a deep learning network leads to sparser representation and more obvious semantics than the pixel space [2, 20, 29], so it might be feasible and potentially beneficial to perform disentanglement in the feature space.



(a) Manipulation of background objects.

(b) Manipulation of OCC-coded patterns.

Figure 4: Background objects and OCC-coded stripes can be roughly disentangled in the latent space.

To verify the feasibility of feature-space disentanglement of OCC-coded pattern and background objects, we adopt a well-known deep learning tool, β-VAE [20], to manipulate the latent variables in the feature space. β-VAE leverages a pair of encoder and decoder to map the input image to a low-dimensional latent representation, and it further achieves statistical independence and disentanglement of the latent variables by imposing a limit on the capacity of the latent information channel. In order to demonstrate that the disentanglement can be achieved successfully, we reconstruct images from the feature space of raw images with mixed background objects and foreground OCC stripes; the results are visualized in Figure 4.

In Figure 4a, the strengths of latent variables controlling the generation of a teapot are gradually decreased, forcing the teapot to gradually fade out in the reconstructed image till leaving only residuals and stripes eventually. Similarly in Figure 4b, the strengths of latent variables controlling the generation of the OCC-coded stripes are gradually increased, enabling the bright-dark stripes to be increasingly introduced into the reconstructed image. The ability to respectively control the proportions of the background objects and OCC-coded pattern indicates that the two components can be roughly disentangled. However, two challenges of this method remain to be tackled. On one hand, as shown in Figure 4a, the background scene cannot be completely removed from the OCC-coded pattern, potentially causing OCC decoding errors. On the other hand, the reconstructed background images significantly lack of details, which in turn affects the performance of OR and general CV functions beyond it (whose success only depends on features). We will rise to these challenges in Section 3.

3 CORE-LENS DESIGN

In this section, we explain the design of CORE-Lens. Starting with a brief overview of the CORE-Lens workflow, we then present how DRL helps separate mixed components in latent space. We further leverage GAN to enhance DRL for generating sharp background images, on which OR and OCC can be performed in a virtually transparent manner.

3.1 Overview

Our goal is to perform OR and OCC simultaneously on smartphones, hence realizing a preliminary ISAC prototype for VLC on smart devices. Although a trivial time-division solution could naturally decouple these two functions, it would significantly reduce the efficiency of both sensing (OR) and communication (OCC); hence we stress the *concurrency* between OR and OCC in our design. Of course, to realize this goal, we face the challenge of mutual interference between the two functions demonstrated in Section 2. Consequently, we propose CORE-Lens to tackle these challenges. Built upon the idea of DRL, CORE-Lens disentangles OCC-coded light and background objects in latent space. The intuition is that, although the two components are mixed and inseparable in the pixel space, their high-level features (e.g., texture and shape) can be captured and disentangled by specially designed deep learning networks. As a result, the separated features would in turn allow us to reconstruct images from them to facilitate respective tasks.

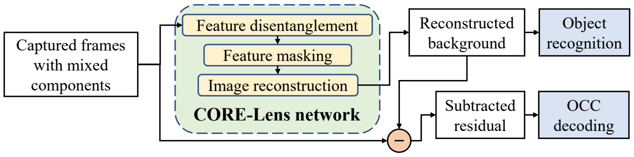


Figure 5: Diagram of CORE-Lens workflow.

Figure 5 illustrates the workflow of CORE-Lens. The frames (or images) captured by a camera contain mixed OCC-coded patterns and background objects. They are fed to *CORE-Lens network* to perform feature disentanglement and separation, and the resulting features are exploited to drive image reconstructions respectively for OR and OCC. More specifically, CORE-Lens leverages an upgraded VAE (variational autoencoder) [27] to generate a latent space with disentangled features. It then employs the self-attention mechanism [54] to focus on the OR-related features while masking irrelevant ones. The network further leverages a modified GAN [17] to reconstruct detailed background from relevant features. While a GAN-reconstructed image can be directly used for OR, subtracting it from the original image yields OCC-coded patterns in the residual to further enable OCC decoding.

3.2 Learning Disentangled Representations

To obtain separable representations of background objects and OCC-coded patterns, we hereby perform feature disentanglement and masking.

3.2.1 Feature Disentanglement. We first explain the process of disentangling the features of the background objects and OCC-coded patterns. Let $\mathbf{x} \in \mathscr{P}$ denote the camera-captured images; they contain the mixture (or overlapping) of background objects $\mathbf{x}_{\text{OR}}$ and OCC-coded patterns $\mathbf{x}_{\text{OCC}}$, together with ambient interference and noise ϵ:

$$\mathbf{x} = \mathbf{x}_{\text{OR}} + \mathbf{x}_{\text{OCC}} + \epsilon. \tag{1}$$

For the three reasons explained in Section 2.4, $\mathbf{x}$ cannot be directly decomposed into $\mathbf{x}_{\text{OR}}$ and $\mathbf{x}_{\text{OCC}}$ in its *pixel space* $\mathscr{P}$. Therefore, it is desirable to infer *latent variables* $\mathbf{z} \in \mathscr{L}$ that characterize all variations in $\mathbf{x}$. Since $\mathbf{z}$ is much sparser and contains far more distinguishable semantics than $\mathbf{x}$, disentangling the two mixed components should be feasible within the *latent space* $\mathscr{L}$.

Mathematically, obtaining the latent variables $\mathbf{z}$ can be formulated as inferring its posterior distribution $p(\mathbf{z}|\mathbf{x})$ given an image $\mathbf{x}$. However, since $p(\mathbf{z}|\mathbf{x})$ is intractable, a Bayesian optimization approach, often known as *variational inference* [3], is thus leveraged to approach the problem. To be specific, a surrogate distribution $q(\mathbf{z})$ is employed to approximate $p(\mathbf{z}|\mathbf{x})$ by minimizing their KL (Kullback–Leibler) divergence [53]. Though the KL divergence itself still involves the intractable posterior $p(\mathbf{z}|\mathbf{x})$, the problem can be readily addressed by further decomposing the KL divergence into:

$$\mathbb{KL}\left(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x})\right) = \log p(\mathbf{x}) - \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right]. \tag{2}$$

Since the marginal log-likelihood $\log p(\mathbf{x})$ is independent of the variational distribution $q(\mathbf{z})$, the KL divergence can be minimized by maximizing the variational lower bound $\mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z})}\right]$. To implement the idea of variational inference, we employ the well-known VAE [27] to act as a generative model, whose encoder-decoder structure is shown by the green boxes in Figure 6.

To further introduce the ability of feature disentanglement, we stress that the distribution q is conditioned on the observation $\mathbf{x}$ and approximated by an encoder network $q_\phi(\mathbf{z}|\mathbf{x})$ parameterized by ϕ. Meanwhile, the likelihood $p(\mathbf{x})$ is approximated by a decoder network $p_\theta(\mathbf{x}|\mathbf{z})$ parameterized by θ. Consequently, the variational lower bound of VAE can be expressed as:

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathbb{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right). \tag{3}$$

where the first term $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]$ can be implemented as MSE (mean-square error) reconstruction loss, and the second term $\mathbb{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right)$ is often realized by the KL divergence between the encoder-generated distribution (parameterized by mean μ and standard deviation Σ) and a standard isotropic Gaussian prior. Suppose the output of the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ is $\mathbf{x}'$, then the VAE loss can be practically implemented as:

$$\mathcal{L}_{\text{VAE}} = \left\|\mathbf{x}-\mathbf{x}'\right\| - \mathbb{KL}\left(\mathcal{N}\left(\boldsymbol{\mu},\Sigma^2\right),\mathcal{N}(\mathbf{0},\mathbf{I})\right). \tag{4}$$

The covariance of the isotropic Gaussian prior being equal to an identity matrix $\mathbf{I}$ implies that all the dimensions of $\mathbf{z}$ are independent and thus disentangled. Therefore, if we emphasize the KL divergence by adding a weight β ($\beta > 1$), feature disentanglement can be achieved [20]:

$$\mathcal{L}_{\text{VAE}} = \left\|\mathbf{x}-\mathbf{x}'\right\| - \beta\mathbb{KL}\left(\mathcal{N}\left(\boldsymbol{\mu},\Sigma^2\right),\mathcal{N}(\mathbf{0},\mathbf{I})\right). \tag{5}$$

The disentanglement of the latent space suggests that the latent representation $\mathbf{z}$ can be potentially represented as two statistically independent groups of latent variables, i.e.,

$$\mathbf{z} = \left(\mathbf{z}_{\text{OR}} \oplus \mathbf{z}_{\text{OCC}}\right), \tag{6}$$

in which $\mathbf{z}_{\text{OR}}$ and $\mathbf{z}_{\text{OCC}}$ characterize background objects $\mathbf{x}_{\text{OR}}$ and OCC-coded patterns $\mathbf{x}_{\text{OCC}}$ in the pixel space, respectively. Though this method appears to be plausible so far, it must rely on manually

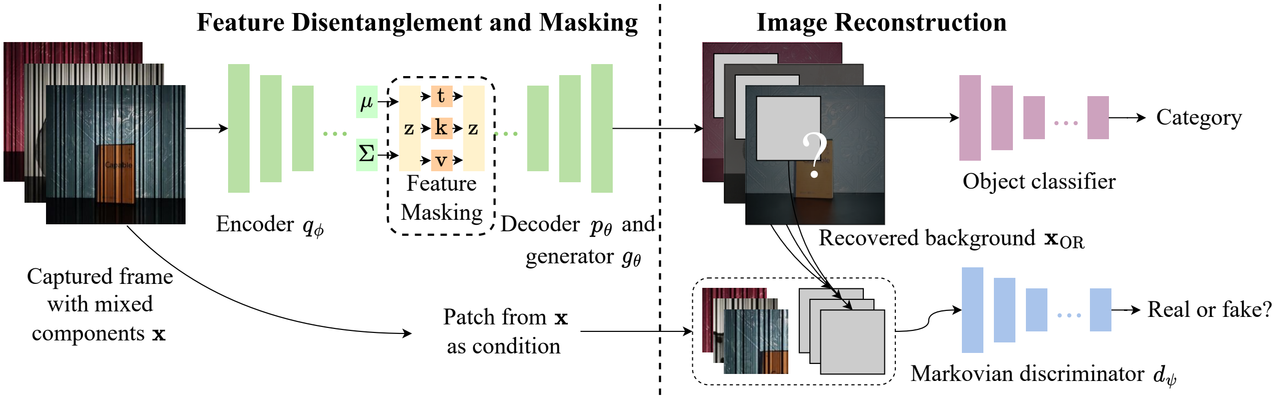

Feature Disentanglement and Masking
Image Reconstruction
μ
Σ
t
z
k
z
v
Feature Masking
Encoder q_ϕ
Decoder p_θ and generator g_θ
Captured frame with mixed components $\mathbf{x}$
Patch from $\mathbf{x}$ as condition
Object classifier
Category
Recovered background $\mathbf{x}_{\text{OR}}$
Real or fake?
Markovian discriminator d_ψ

Figure 6: CORE-Lens network: a combination of variational inference and conditional adversarial learning. Since the (pre-trained) object classifier is not part of the CORE-Lens pipeline, its parameters are not tuned during training.

tuning individual variables in $\mathbf{z}$ to achieve intended separation. Therefore, we need a more effective way to automatically retrieve either $\mathbf{z}_{\text{OR}}$ or $\mathbf{z}_{\text{OCC}}$ according to application requirements.

3.2.2 Feature Masking. Although $\mathbf{z}_{\text{OR}}$ and $\mathbf{z}_{\text{OCC}}$ are disentangled in the latent space $\mathscr{L}$ via our upgraded VAE, their exact correspondences with their counterparts in the pixel space $\mathscr{P}$ are unknown; in other words, we have no idea about which latent variables are responsible for respectively generating background objects $\mathbf{x}_{\text{OR}}$ and OCC patterns $\mathbf{x}_{\text{OCC}}$. To this end, we employ the attention mechanism [54] for automatically selecting relevant features for image reconstruction while masking irrelevant ones. The idea behind the attention mechanism is a selective adoption of the most relevant parts of latent representation $\mathbf{z}$ in a flexible manner, by learning a weighted combination of all variables in $\mathbf{z}$ so that the most relevant variables are given the highest weights.

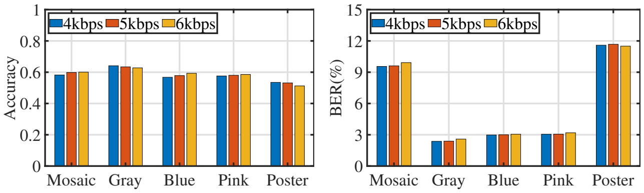
Essentially, the attention function can be described as transforming the latent representation $\mathbf{z}$ to a query $\mathbf{t}$ and a set of key-value pair $\mathbf{k}$ and $\mathbf{v}$, and then mapping them to an output. The query, keys, and values are all linearly transformed versions of the input $\mathbf{z}$:

$$\mathbf{t} = \mathbf{W}_t\mathbf{z} + \mathbf{b}_t,\ \ \mathbf{k} = \mathbf{W}_k\mathbf{z} + \mathbf{b}_k,\ \ \mathbf{v} = \mathbf{W}_v\mathbf{z} + \mathbf{b}_v, \tag{7}$$

where $\mathbf{W}_t, \mathbf{W}_k, \mathbf{W}_v$ and $\mathbf{b}_t, \mathbf{b}_k, \mathbf{b}_v$ are trainable matrices and vectors that help transforming the input to its corresponding query $\mathbf{t}$, key $\mathbf{k}$, and value $\mathbf{v}$, whose dimensions are denoted by d_q, d_k, d_v, respectively. The output context $\mathbf{z}'$ is obtained as a weighted sum of the values in $\mathbf{v}$, where the weight of each value is a normalized product of the query $\mathbf{t}$ and its corresponding key $\mathbf{k}$: $\mathbf{z}' = \text{softmax}\left(\frac{1}{\sqrt{d_k}}\mathbf{t}\mathbf{k}^T\right)\mathbf{v}$. The working principle of attention is briefly illustrated by the "Feature Masking" module in Figure 6. After being properly trained, the attention mechanism acts as a "mask" to help focusing on either $\mathbf{z}_{\text{OR}}$ or $\mathbf{z}_{\text{OCC}}$, thus facilitating recovering $\mathbf{x}_{\text{OR}}$ or $\mathbf{x}_{\text{OCC}}$, respectively.

3.3 Image Reconstruction

With the preparation of the last two steps, we have obtained a disentangled and masked latent representation $\mathbf{z}'$. Now the task becomes how to reconstruct the background and OCC patterns from $\mathbf{z}'$. Although the VAE decoder discussed in Section 3.2.1 does yield images containing a disentangled component, such images can be very blurry as demonstrated in Figure 4. If we perform OR or OCC on these recovered images, the results shown in Figure 7 indicate the average OR accuracy and OCC BER as 60% and 6%, respectively; both are barely usable. In fact, it is theoretically explainable why common VAEs often fail to render sharp images: the element-wise MSE loss in Eqn. (5), though simple to implement, cannot model high-level (abstract) features.[2] In order to generate sharper images, we require a new approach that measures the reconstruction performance based on more abstract features.



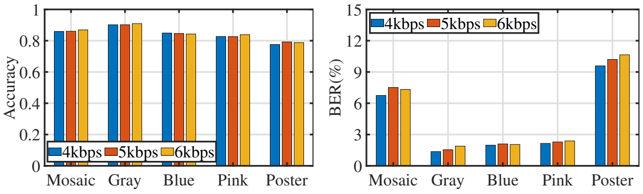
(a) OR accuracy. **(b) OCC BER.**

Figure 7: Images reconstructed by VAE decoder offer limited OR and OCC performance.

Rather than handcrafting high-level features to measure the reconstruction performance, we employ a specially designed cGAN (conditional generative adversarial network) to automatically learn them. Our CORE-Lens cGAN consists of a generator g_θ and a Markovian discriminator [24] d_ψ parameterized by θ and ψ, respectively. The generator g_θ takes in both the captured image $\mathbf{x}$ and the disentangled representation $\mathbf{z}'$ to reconstruct the background objects and OCC patterns, and the Markovian discriminator d_ψ distinguishes images produced by the generator from the true data distribution, but only penalizes structure at the scale of patches.[3] The cGAN objective is i) to find d_ψ that gives the best possible discrimination between true and reconstructed images, and ii) to encourage g_θ to

[2] *High-level* features are those composed of many *low-level* features (e.g., edge, shape, and texture); they often represent abstract ideas, such as the existence of certain objects or if an image is fake or not.

[3] d_ψ models the image as a Markovian random field [24], assuming independence between pixels separated by more than the diameter of a patch.



(a) OR accuracy. **(b) OCC BER.**

Figure 8: VAE+cGAN reconstructed background images lead to an adequate OR accuracy, but similarly reconstructed OCC patterns cannot yield a low BER.

fit the true data distribution offered by the prior $\mathbf{x}$. Therefore, the loss function of cGAN can be represented as a binary cross entropy:

$$\mathcal{L}_{\text{cGAN}} = \log\left(d_\psi(\mathbf{x}_{\mathbf{OR}},\mathbf{x})\right) + \log\left(1 - d_\psi(g_\theta(\mathbf{z}'),\mathbf{x})\right). \qquad (8)$$

In fact, we can merge the VAE decoder p_θ and the cGAN generator g_θ into one by sharing their parameters. Moreover, we can train the VAE and cGAN jointly by combining their loss functions. Now CORE-Lens has the best of both VAE and cGAN, thus it should recover a much sharper image than the original VAE network. We employ CORE-Lens to reconstruct both background objects and OCC-coded patterns, and show their corresponding OR accuracy and OCC BER in Figure 8. One may readily observe that the OR accuracy is greatly improved over the images generated by VAE. Nonetheless, the worst-case BER of OCC, at around 10%, is too high to be usable for communication purposes.

We attribute this unsatisfactory OCC performance to the blurred edges in the reconstructed OCC-coded patterns: although the patterns generated by CORE-Lens are much sharper than those generated by VAE, they are still insufficient for decoding purposes. Fortunately, we notice that the OCC patterns in the original mixture are sharp, motivating an alternative to subtract the reconstructed background $\mathbf{x}'_{\text{OR}}$ from to the original mixture $\mathbf{x}$ to approximate OCC-coded patterns $\mathbf{x}_{\text{OCC}}$. To guarantee that the residual OCC patterns are accurate, we modify the VAE loss in Eqn. (5) as:

$$\begin{aligned}\mathcal{L}_{\text{VAE}} \;=\;& \left\|\mathbf{x}_{\text{OR}} - \mathbf{x}'_{\text{OR}}\right\| + \left\|\hat{\mathbf{x}}_{\text{OCC}} - (\mathbf{x} - \mathbf{x}'_{\text{OR}})\right\| \\ -\;& \beta\mathbb{KL}\left(\mathcal{N}\left(\boldsymbol{\mu},\Sigma^2\right),\mathcal{N}(\mathbf{0},\mathbf{I})\right), \qquad (9)\end{aligned}$$

where $\hat{\mathbf{x}}_{\text{OCC}}$ is OCC-coded patterns directly generated from OCC bitstream. To summarize, the overall loss function of CORE-Lens becomes:

$$\begin{aligned}\mathcal{L}_{\text{CORE-Lens}} &= \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{cGAN}} \\ &= \left\|\mathbf{x}_{\text{OR}} - \mathbf{x}'_{\text{OR}}\right\| + \left\|\hat{\mathbf{x}}_{\text{OCC}} - (\mathbf{x} - \mathbf{x}'_{\text{OR}})\right\| \\ &\quad - \beta\mathbb{KL}\left(\mathcal{N}\left(\boldsymbol{\mu},\Sigma^2\right),\mathcal{N}(\mathbf{0},\mathbf{I})\right) \\ &\quad + \log(d_\psi(\mathbf{x}_{\mathbf{OR}},\mathbf{x})) + \log(1 - d_\psi(g_\theta(\mathbf{z}'),\mathbf{x})). \qquad (10)\end{aligned}$$

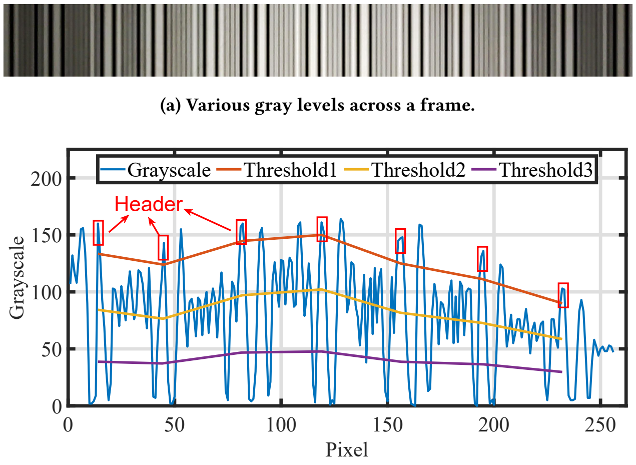
The overall CORE-Lens design guarantees that the reconstructed background image $\mathbf{x}'_{\text{OR}}$ is sharp and accurate for OR. Meanwhile, the residuals $\mathbf{x}'_{\text{OCC}}$ obtained by subtracting the backgrounds from the captured images $\mathbf{x}$ preserve OCC information and are largely free of interference.

3.4 OCC Decoding on Residual Frame

Though CORE-Lens has managed to largely separate the mixture $\mathbf{x}$ into $\mathbf{x}'_{\text{OR}}$ and $\mathbf{x}'_{\text{OCC}}$, the results cannot be as clear as they were separately captured in time. It is true that $\mathbf{x}'_{\text{OR}}$ can be directly taken by an OR network (e.g., [19, 22, 50]) to perform classification, $\mathbf{x}'_{\text{OCC}}$ may still lead to non-negligible BER, albeit being further reduced compared with that in Figure 8b. To cope with this situation, we add another pre-processing component before an arbitrary OCC decoder, though we adopt ReflexCode [65] as the example decoder in the following descriptions thanks to its code availability. Also, we stick to OOK (on-off keying), the most robust OCC modulation, for our initial study on ISAC-OCC, as applying higher-order modulations (albeit raising bit rate) can significantly increase BER.

The OCC data are encoded by the Manchester OOK, a robust modulation scheme. In the case of low data rate communication, the Manchester OOK demodulation needs to distinguish only two gray levels: ON (or "1") and OFF (or "0"). However, in cases of high data rate, the combined effect of the rolling shutter and limited responsiveness of an image sensor results in the emergence of various gray levels [13], and this situation can be exacerbated by CORE-Lens reconstruction. Specifically, there could be at least four gray levels corresponding to "00", "01", "10", and "11" in ascending order. Figure 9a shows a Manchester OOK example of various gray levels under white background. As Manchester modulation mandates the maximum sequence of similar symbols to have a length of two, we leverage this to identify the error bits in the received data.

To retrieve the information contained in $\mathbf{x}'_{\text{OCC}}$, we first convert the 2-D OCC-coded patterns into 1-D OCC-coded sequences. To be specific, we obtain a sequence with maximal SNR (signal-noise-ratio) by leveraging frame-averaging [35] on $\mathbf{x}'_{\text{OCC}}$. Nonetheless, the sequence may still not be readily decodable due to the non-uniformity of grayscale induced by residual interference from ambient light and background. To give an example, the grayscale shown in Figure 9b does not have a monotonic trend; it is hence impractical to use straightforward thresholds for decoding the grayscale sequence. To tackle this challenge, we leverage the fact that the



(a) Various gray levels across a frame.

(b) Illustration of the demodulation procedure.

Figure 9: Demodulation of OCC bits coded by Manchester OOK under high data rate scenarios.

sequence is approximately coherent (i.e., can be deemed monotonic) between two neighboring headers to obtain adaptive thresholds. Specifically, we first locate the headers by their periodicity and local maxima in a packet, as shown in the red rectangular box in Figure 9b. Once all headers are located, we use the header's grayscale to determine a piecewise linear function as the thresholds (red, yellow, and purple lines in Figure 9b), whose levels are empirically determined so as to minimize BER. Subsequently, the bits between two adjacent headers can be demodulated/decoded by an arbitrary OCC decoder, resulting in a complete decoding of a whole frame.

4 IMPLEMENTATIONS AND DATASET

In this section, we first introduce both hardware and software implementations of CORE-Lens, then explain how our dataset is collected and processed.

4.1 Prototype and Experiment Setup

To implement the CORE-Lens prototype, we employ a 12 W LED spotlight as the luminaire to emulate the reflected lighting in a museum display setting. The LED driver circuits consist of an AC-DC converter, a microcontroller, and a few MOSFETs. To be specific, the AC-DC converter powers the spotlight by taking 220 V AC input and outputting 40 V DC, the SI2310A MOSFET [52] amplifies the modulated signal and directly drives the spotlight; it is in turn controlled by an ARM Cortex-M4 GD32F330G8U6 microcontroller [16] to perform modulations. For the receiving camera, we test ten different smartphones but report only the results of Huawei Mate 30 Pro [23]: since CORE-Lens needs to be retrained for each phone model, the performance of both OR and OCC can be made largely insensitive to phone models as far as they are sufficiently powerful. The CORE-Lens prototype (notable the app interface) and experiment setup are shown in Figure 10.

The software implementation of CORE-Lens is based on Python 3.7, with the deep learning network and OCC decoding module built upon PyTorch 1.7.1 [45] and OpenCV [5], respectively. The encoder q_ϕ consists of repeated units of two convolutional layers of kernel size 3, each followed by a ReLU (rectified linear unit) and a max-pooling layer with stride 2 for downsampling; the number of feature channels is doubled at each downsampling step. The two-layer convolutional units are repeated 4 times in the encoder. The generator g_θ consists of repeated units of two convolutional layers of kernel size 3, each followed by a transposed convolution layer of kernel size 2 and a ReLU layer. The transposed convolutional layer upsamples the feature map and halves the number of feature channels. Similar to the encoder, the two-layer convolutional units are repeated 4 times in the decoder. The Markovian discriminator d_ψ is implemented as a four-layer fully convolutional network with a perceptive field of 30.

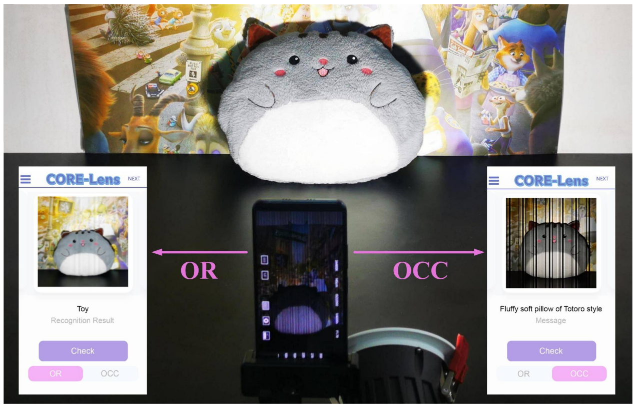


Figure 10: Illustration of CORE-Lens prototype (partial for the LED lumimaire) and experiment setup.

4.2 Dataset and Network Training

Since there is no publicly available dataset for evaluation of simultaneous OR and OCC, we collect and prepare our own. Specifically, the dataset is collected under normal ambient lighting, by illuminating OCC-coded light upon different backgrounds containing 10 classes of objects, namely book (BK), bottle (BL), box (BX), scissor (SS), bag (BG), laptop (LT), toy (TY), teapot (TT), plant (PT), and cap (CP), put in front of 6 different scenes (namely white wall, mosaic, gray, blue, pink and poster). To produce diversified OCC-coded patterns, we transmit random data in 5 packet formats with different headers and packet lengths. Each collected frame is resized to 256 × 256 pixels to fit the CORE-Lens network. We collect 20,000 frames in total, including 16,000 frames for training CORE-Lens and 4,000 for testing, which is made public at [14] and will be further enriched; some example frames in our dataset are shown in Figure 11. As the dataset includes frames with diversified domain information (e.g., scene, distance, orientation, and ambient illumination), we also verify the cross-domain generalization capability

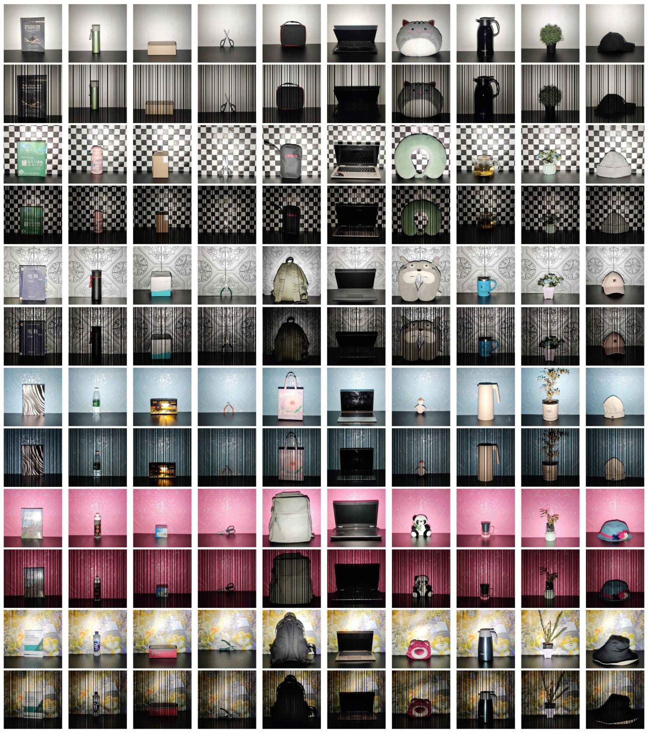


Figure 11: Examples of captured frames in our dataset.

of CORE-Lens by separating the training and testing domains. For the training process, all weights in the CORE-Lens network are initialized by the Xavier uniform initializer [30], and the batch size is set to 64, the loss in Eqn. (10) is adopted with the weight β set to 3. The learning rate and momentum of the SGD optimizer [4] are set to 0.01 and 0.9, respectively. The CORE-Lens network is trained for 1000 epochs on an NVIDIA GeForce RTX 2070 SUPER GPU [42], and the training process costs 5 hours in total. To perform inference on mobile devices, we port the trained network to Android and iOS platforms by Pytorch Mobile [46].

5 EVALUATION

In this section, we evaluate both OR and OCC performance of CORE-Lens under various experiment setups, and we also discuss features and limitations common to both OR and OCC functions.

5.1 OR Performance

We start with evaluating the OR performance. We first explain the reason for the improved OR performance by visualizing the saliency maps obtained by CORE-Lens. We further study how the increasing class cardinality, varying data rates, and different backgrounds impact the OR performance, adopting CORE-Lens network without the discriminator d_ψ as the baseline classifier for comparison purposes, as otherwise commonly used baselines (e.g., GoogLeNet [50]) perform much worse (see Section 2.2). Due to page limit, we directly adopt optimized hyperparameters, e.g., β and learning rate, but omit their empirical evaluations.

5.1.1 Saliency Map Visualization. A saliency map is an image in which the grayscale values of the pixels are marked according to their contribution to the object recognition task [48]. By creating a saliency map for CORE-Lens, we can gain intuition on where the network is paying the most attention to in an input image. Figure 12a shows a captured image of a pot of plant interfered with OCC-coded patterns. We employ a Python library FlashTorch [43] to visualize its saliency maps and show the results of the baseline and CORE-Lens network in Figures 12b and 12c, respectively. One may readily observe that the saliency map of the baseline classifier is diffusive, i.e., the attention is shifted towards abnormal OCC-coded patterns instead of being stressed on the real object. As a comparison, the saliency map of CORE-Lens focuses mostly on the object barely affected by the OCC-coded patterns. The distinct difference evidently demonstrates the disentanglement capability of CORE-Lens, which in turn leads to then enhanced OR performance.

5.1.2 Overall Performance. We first discuss the overall OR performance of CORE-Lens shown in Figure 13. Basically, we consider three OCC data rates (4, 5, and 6 kbps), and for each data rate, we first look at how OR accuracy relates with the number of object classes, then we conduct a detailed inspection on the confusion matrices that especially reflect the OR accuracy given all 10 object classes. It can be observed, from top panels, that the OR accuracy is 96% on average when the received frames are clean and free from OCC interference. Although this accuracy slightly drops to 92% when there exists OCC interference, it is already a huge improvement over the baseline, whose OR accuracy for OCC interfered frames may drop below 70% eventually. Moreover, the OR accuracy (for both clean and OCC interfered frames) is shown to be largely insensitive to varying data rates given the full CORE-Lens network, demonstrating one aspect of the robustness of CORE-Lens.

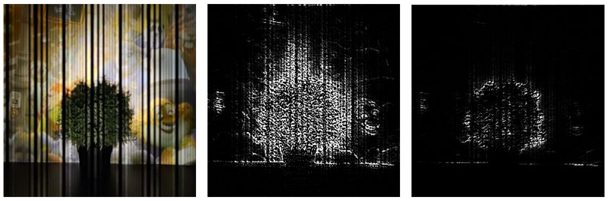
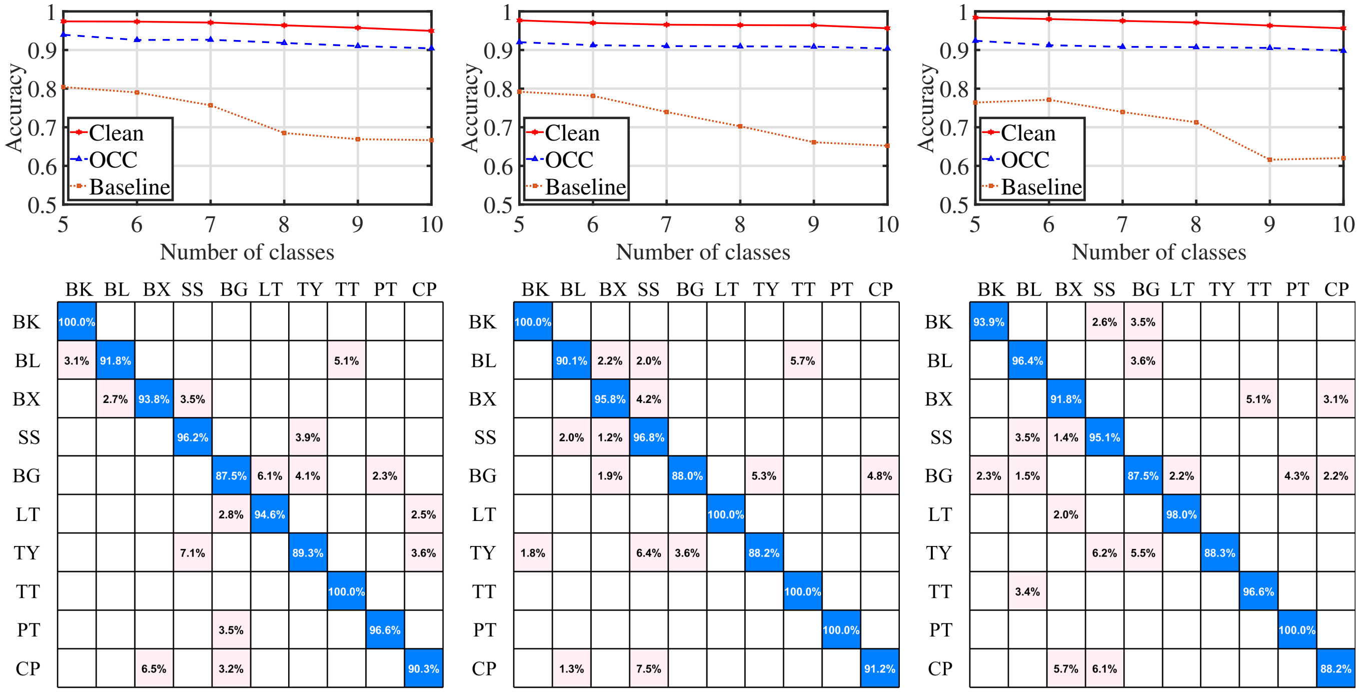

(a) Raw. (b) Baseline. (c) CORE-Lens.

Figure 12: Saliency maps of a pot of plant.

We then inspect OR performance under different numbers of classes. The top panel of Figure 13 shows that the OR accuracy of CORE-Lens is fairly stable from 5 to 10 classes, and the overall variation of OR accuracy does not exceed 4%. This stability is in stark contrast to the baseline, whose accuracy drops by 12%, 15%, and 18% when the number of classes increases from 5 to 10 and the data rate is at 4kbps, 5kbps, and 6kbps, respectively. The results suggest that a VAE alone in the baseline network, though outperforming normal CNN classifiers such as GoogLeNet [50], is still insufficient for disentangling background objects and OCC-coded patterns, whereas the Markovian discriminator d_ψ can help CORE-Lens enhancing its disentanglement capability and generating clearer pictures as well, thus remaining robust in the face of an increasing class cardinality.

We further explore the detailed OR performance of CORE-Lens under all 10 object classes and three data rates using confusion matrices. From the bottom panel of Figure 13, we can observe that the OR accuracy of almost all classes stays beyond 90% (some even get close or reach 100%), except the “BG” (bag), “TY” (toy), and “CP” (cap). By inspecting our dataset (as illustrated by Figure 11), we conjecture that the possible reason is that these classes are larger in shapes and span across more non-uniform OCC-coded patterns, thus receiving more interference. Additionally, we find that misclassified objects are most likely to fall into “SS” (scissors), with an average misclassification rate (over all classes) of 1.06%, 2.01%, and 1.49% under data rates of 4 kbps, 5 kbps, and 6 kbps, respectively. This may be explained by the fact that scissors are inconspicuous by occupying a small portion of a frame, thus leading to relatively weak activations in CORE-Lens network and tending to be mistakenly imitated by objects from other classes.

5.1.3 Impact of Varying Backgrounds. We further evaluate the OR performance of CORE-Lens under different wall backgrounds (e.g., white, mosaic, and colored walls, as well as colorful posters) with a fixed 5 kbps data rate. The OR performance of CORE-Lens is compared with the baseline and reported in Figure 14. Figure 14a shows that the OR accuracy of CORE-Lens on clean frames has a narrow improvement of around 4% over the baseline classifier, whereas Figure 14b demonstrates that CORE-Lens is far superior to the baseline with more than 20% increase in OR accuracy when captured frames having varying backgrounds. This conspicuous improvement in accuracy from the baseline to CORE-Lens confirms that CORE-Lens is largely immune to excessive interference caused by varying backgrounds, thanks to its capability in disentangling the mixed signal features in captured OCC frames.



(a) 4 kbps

	BK	BL	BX	SS	BG	LT	TY	TT	PT	CP
BK	100.0%									
BL	3.1%	91.8%						5.1%		
BX		2.7%	93.8%	3.5%						
SS				96.2%			3.9%			
BG					87.5%	6.1%	4.1%		2.3%	
LT					2.8%	94.6%				2.5%
TY				7.1%			89.3%			3.6%
TT								100.0%		
PT					3.5%				96.6%	
CP			6.5%		3.2%					90.3%

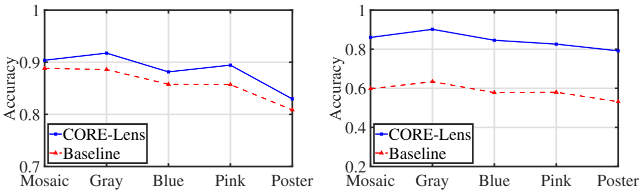
(b) 5 kbps

	BK	BL	BX	SS	BG	LT	TY	TT	PT	CP
BK	100.0%									
BL		90.1%	2.2%	2.0%				5.7%		
BX			95.8%	4.2%						
SS		2.0%	1.2%	96.8%						
BG			1.9%		88.0%		5.3%			4.8%
LT						100.0%				
TY	1.8%			6.4%	3.6%		88.2%			
TT								100.0%		
PT									100.0%	
CP		1.3%		7.5%						91.2%

(c) 6 kbps

	BK	BL	BX	SS	BG	LT	TY	TT	PT	CP
BK	93.9%			2.6%	3.5%					
BL		96.4%			3.6%					
BX			91.8%					5.1%		3.1%
SS		3.5%	1.4%	95.1%						
BG	2.3%	1.5%			87.5%	2.2%			4.3%	2.2%
LT			2.0%			98.0%				
TY				6.2%	5.5%		88.3%			
TT		3.4%						96.6%		
PT									100.0%	
CP			5.7%	6.1%						88.2%

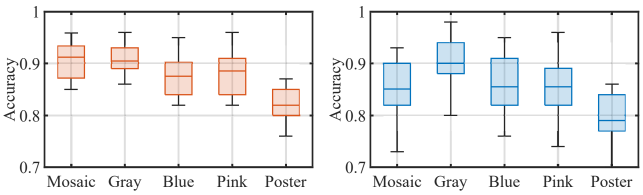
Figure 13: The OR performance under an increasing class cardinality and varying data rate, as well as the confusion matrices for all 10 object classes of objects, both given a white wall as the background.



(a) Clean frames. **(b) OCC frames.**

Figure 14: OR performance of CORE-Lens and baseline classifier under varying backgrounds.

To further inspect CORE-Lens' OR performance under various backgrounds, we use boxplots of OR accuracy to reflect its distribution over 10 object classes. The results on clean and OCC-interfered frames are shown in Figures 15a and 15b, respectively. One may observe that the OR accuracy on OCC-interfered frames has a more spread distribution than that on clean frames, potentially attributed to the unpredictable interference left by OCC-coded patterns on the disentangled background. Furthermore, we notice that the mosaic and poster backgrounds lead to large discrepancies across object classes in OR performance, probably due to their distinct interactions with certain object classes.



(a) Clean frames. **(b) OCC frames.**

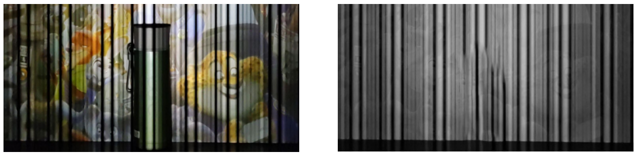
Figure 15: OR performance of CORE-Lens under varying backgrounds.

5.2 OCC Performance

We then evaluate the OCC performance of CORE-Lens. To be specific, we first visualize the reconstructed residual frame $\mathbf{x}'_{\text{OCC}}$ to understand why CORE-Lens improves OCC performance, then study how varying data rates and backgrounds impact OCC performance. We adopt vanilla OCC decoding on raw received frames as the comparison baseline.

5.2.1 Residual Frame. We showcase a captured frame $\mathbf{x}$ and its residual frame $\mathbf{x}'_{\text{OCC}}$ in Figure 16. By subtracting the reconstructed background $\mathbf{x}'_{\text{OR}}$, the object (bottle) and wall patterns in $\mathbf{x}$ have been largely eliminated in $\mathbf{x}'_{\text{OCC}}$. Although there are still minor mismatches and background shades left in $\mathbf{x}'_{\text{OCC}}$, they generally do not degrade the OCC performance of CORE-Lens, as demonstrated in the following experiments, because they are readily handled by the frame-averaging method mentioned in Section 3.4.

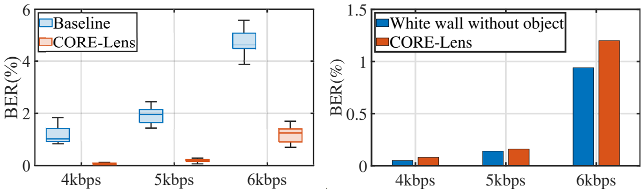
5.2.2 Impact of Varying Data Rates. We compare the BER of CORE-Lens with the baseline method under three data rates in Figure 17a. Apparently, CORE-Lens substantially outperforms the baseline with its median BERs at 0.1%, 0.2%, and 1.3% respectively under data rates of 4 kbps, 5 kbps, and 6 kbps, as the corresponding median



(a) Captured frame x. (b) Residual frame $\mathbf{x}'_{\text{OCC}}$.

Figure 16: From a captured frame (a) to a decodable residual frame (b).

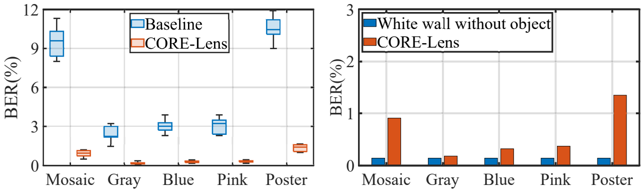
BERs of the baseline is one order higher at 1.6%, 2%, and 4.7%, respectively. This comparison evidently confirms that the baseline method can be severely affected by the background interference, so that communication quality is seriously compromised. As shown in Figure 17b, the OCC performance of CORE-Lens is nearly the same as that evaluated given a white wall background, concretely proving the efficacy of our feature disentangling method in maintaining a reliable OCC channel. We notice that the BER of CORE-Lens experiences a substantial increase to 1.2% when the data rate reaches 6kbps. This phenomenon can be partially explained by the fact that a high data rate significantly complicates OCC demodulation by shrinking the width of OCC-coded stripes. Fortunately, a data rate of 5kbps is often sufficient for most OCC applications [47], so we leave a high-rate ISAC-OCC to future work.



(a) CORE-Lens vs baseline. (b) CORE-Lens vs clean OCC.

Figure 17: OCC performance under varying data rates.

5.2.3 Impact of Varying Backgrounds. We then fix the data rate at 5kbps and evaluate the communication performance of CORE-Lens under varying background scenes, and the results are shown in Figure 18. As expected, the baseline is greatly affected by interference from varying scenes: it has BERs varying from 1.5% to 4% given pure-colored scenes, otherwise it BERs can go up to 12% given more complicated scenes composed of mosaic and colorful posters. As a comparison, CORE-Lens effectively handles the interference and achieves BERs below 1% under all scenes except for the poster, where a slightly higher median BER of 1.3% is reached. As shown in Figure 18b, the BERs achieved by CORE-Lens on varying scenes are close to the ideal case where reflected OCC is performed on white wall scene, again providing the efficacy of CORE-Lens.



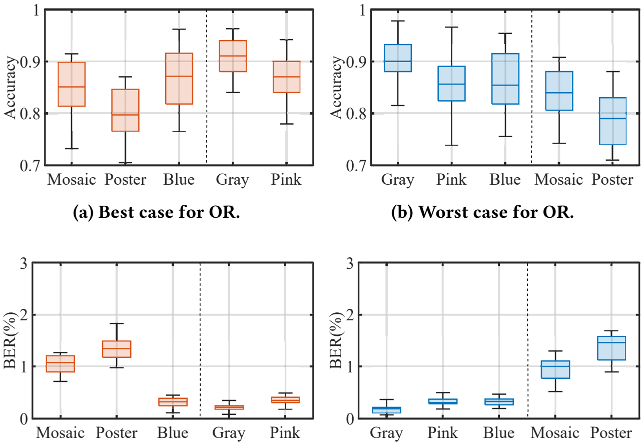
(a) CORE-Lens vs baseline. (b) CORE-Lens vs clean OCC.

Figure 18: OCC performance under varying scenes.

5.3 Cross-Domain Performance of CORE-Lens

Our experiments so far have focused only on the two key metrics, namely OR accuracy and OCC quality, leaving the domain information (scene, distance, orientation, and ambient illumination) blended in the training and testing phases. Therefore, we hereby evaluate the cross-domain performance of CORE-Lens. As it is a convention to normalize frames with respect to both orientation and ambient illumination, we only consider two domain aspects: namely scene and distance.

5.3.1 Cross-Scene. As we have five different scenes (other than the trivial white wall) involved in our dataset, we use three of them to train CORE-Lens and then test the resulted model on the remaining two. Due to the space limit, we only present the results for the best and worst case combinations in Figure 19: the best case uses "difficult" scenes to train (Figures 19a and 19c), while the worst case perform test on them (Figures 19b and 19d). Apparently, training CORE-Lens with the difficult scenes results in very good performance on unseen scenes (which may almost match those reported earlier with mixed-scene training), but testing on these difficult scenes, as expected, leads to relatively worse performance. Since the performance degradation under cross-scene testing most appears to OR, we believe that it can mostly be attributed to the trained OR classifier (not part of our CORE-Lens): OR under dazzling background scenes is known to be a hard problem [39, 58].

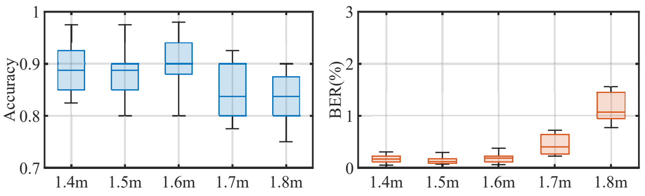


(a) Best case for OR. (b) Worst case for OR.

(c) Best case for OCC. (d) Worst case for OCC.

Figure 19: Cross-scene generalizability of CORE-Lens.

5.3.2 Cross-Distance. Whereas the our CORE-Lens model is trained at a fixed distance of 1.6 m, we hereby demonstrate that the model can be generalize to other distances varying from 1.4 m to 1.8 m. The reason for this relatively narrow range in distance is twofold: i) subject to the nature of reflected OCC (also confined by the adopted LED luminaire), decoding error can increase significantly after the wall-camera distance grows beyond 1.8 m, and ii) once the object-camera distance is reduced below 1.4 m, some scene settings may fall outside the field of view of the camera and cannot be fully



(a) OR.

(b) OCC.

Figure 20: Cross-distance generalizability of CORE-Lens.

captured, thus unfairly affecting the OR accuracy. As shown in Figure 20, the general trend of OR accuracy is slightly decreasing in distance and that of OCC BER is slightly increasing, which is very much expectable given the information theoretical principle that a higher SNR (at a lower distance) should lead to better performance in both OR and OCC. Of course, the generalizability issue of CORE-Lens (in terms of OR) does manifest itself at shorter distances (at longer distances too but not very discernible): it appears that the higher SNR is somehow offset by the amplified image compared those taken at the trained distance. Fortunately, this issue does not appear to be very serious, and it can be handled by normalizing the frame size for all distances.

5.4 Discussions on Limitations

As we explained in Section 1, separating CV and OCC in a time-divided manner is feasible but not efficient. During our experiments, we are able to collect the nominal runtime complexities (i.e., latency in computation) of individual components (as shown in Table 1); they allow us to verify our earlier statement. Taking the Android phone as an example and assuming a tight time-divided schedule to separate OR and OCC, the OR time would be 42.477 ms (with GoogLeNet) while that of OCC alone (with ad hoc filtering) would become 57.511 ms. However, there would be an additional switching time (< 1 ms) for changing the camera exposure parameters (as a sufficiently long exposure could get rid of OCC patterns) and the actual exposure time (1/30 s or 33.333 ms to achieve its purpose). As a result, the whole cycle (one OR and one OCC) takes more than 130 ms. On the contrary, CORE-Lens would only require slightly more than $20.436 + 42.477 = 62.913$ ms, as the processing of OR and OCC can be largely conducted in parallel. Therefore, using CORE-Lens should improve the time efficiency by more than 100%. In addition, adopting the time-divided approach demands a frequent switching between distinct exposure parameters, which should not do any good to the lifetime of the embedded cameras.

Table 1: Latency and memory usage of CORE-Lens and other standalone modules.

	Latency on PC (ms)	Latency on Android (ms)	Latency on iOS (ms)	Memory (MB)
CORE-Lens	3.040	20.436	13.248	68.312
OR	3.835	42.477	35.107	138.489
OCC	1.045	8.072	2.896	14.574
Exposure	–	33.333	33.333	–
OCC w/ filter	4.028	57.511	52.136	18.880

Another concern may come from the real-time OCC performance, which we could also clarify using the data in Table 1. According to Table 1 and taking the Android phone as an example, we should have a frame rate of 15 fps even if OR is performed for every frame, as one cycle of CORE-Lens is about 63 ms based on our earlier calculation. Since each frame consists of four 30-bit packets, the upper bound of the achievable data rate is $30 \times 4 \times 15 = 1.8$ kpbs, largely satisfying the requirements of normal OCC applications. In fact, OR does not have to be performed for each frame in practice, so the data rate can be further improved by allocating more time on OCC. Suppose we perform OR once per second (still a very aggressive setting), there are $1000 - 63 = 937$ ms left for OCC decoding, readily supporting a frame rate of $937 \div 29 = 32$ fps. Since CORE-Lens is the first prototype built for realizing ISAC-OCC, there should be plenty of room for further optimizing the runtime complexity in future developments for specific applications.

One limitation of CORE-Lens in terms of CV functions is that it can only support "macro" tasks such as OR but not "micro" ones such as human activity recognition [36, 38], position/orientation tracking [41], and remote vital signs monitoring [8, 44, 66], as the reconstructed frames may not retain all the necessary details. Although researchers can leverage other sensing media (e.g., [6, 9, 10, 15, 67]) to perform such tasks in an independent manner, it is still our future goal to merge OCC with general CV under the ISAC framework. The other limitation has to do with the modulation adopted by CORE-Lens: the current implementation is based only on OOK, the most basic modulation, for its robustness. Therefore, it remains to be challenging to apply high-rate (higher-order) modulations to CORE-Lens. Last but not least, though pioneering the ISAC construction in the visible light regime, CORE-Lens is still confined to camera-based OCC, so we are on the way towards exploring ISAC under general VLC settings.

6 CONCLUSION

Simultaneous communication and sensing is a challenging yet important problem for visible light related applications, because it unlocks the full potential of widely adopted LED lighting infrastructures and embedded cameras. To this end, we have implemented CORE-Lens, a smartphone-based solution that performs both communication (OCC) and sensing (OR in particular) currently. Employing carefully designed deep learning modules, CORE-Lens first leverages disentangled representation learning to separate the background and OCC-coded patterns in the feature space, and then performs robust OR and OCC on the GAN-recovered frames, respectively. With extensive experiments under different data rates and background objects, we have demonstrated the promising performance of CORE-Lens in simultaneous OR and OCC. We are actively seeking relevant application scenarios for deploying CORE-Lens, in order to promote a wider acceptance of our ISAC-VLC concept.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their valuable and constructive comments. This work was supported in part by National Natural Science Foundation of China under Grant No. 61902267, 62101359 and 62072319. We also thank Energy Research Institute @ NTU (ERI@N) and NTU-IGP for the PhD scholarship of Tianyue Zheng. Yanbing Yang is the corresponding author.

REFERENCES

[1] Navid Bani Hassan, Zabih Ghassemlooy, Stanislav Zvanovec, Mauro Biagi, Anna Maria Vegni, Min Zhang, and Pengfei Luo. 2019. Non-Line-of-Sight MIMO Space-Time Division Multiplexing Visible Light Optical Camera Communications. *OSA/IEEE Journal of Lightwave Technology* 37, 10 (2019), 2409–2417.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

[3] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational Inference: A Review for Statisticians. *Journal of the American statistical Association* 112, 518 (2017), 859–877.

[4] Léon Bottou. 2012. Stochastic Gradient Descent Tricks. In *Neural Networks: Tricks of the Trade*. Springer, 421–436.

[5] Gary Bradski and Adrian Kaehler. 2000. OpenCV. *Dr. Dobb's Journal of Software Tools* 3 (2000), 2.

[6] Chao Cai, Rong Zheng, and Jun Luo. 2022. Ubiquitous Acoustic Sensing on Commodity IoT Devices: A Survey. *IEEE Communications Surveys & Tutorials* 24, 1 (2022), 432–454.

[7] Shao-Qi Chen, Xue-Fen Chi, and Te-Yu Li. 2021. Non-line-of-sight Optical Camera Communication Aided by a Pilot. *Opt. Lett.* 46, 14 (2021), 3348–3351.

[8] Weixuan Chen and Daniel McDuff. 2018. DeepPhys: Video-Based Physiological Measurement Using Convolutional Attention Networks. In *Proc. of the 15th IEEE ECCV*. 349–365.

[9] Zhe Chen, Chao Cai, Tianyue Zheng, Jun Luo, Jie Xiong, and Xin Wang. 2021. RF-Based Human Activity Recognition Using Signal Adapted Convolutional Neural Network. *IEEE Trans. on Mobile Computing* (2021), 1–13.

[10] Zhe Chen, Tianyue Zheng, Chao Cai, and Jun Luo. 2021. MoVi-Fi: Motion-robust Vital Signs Waveform Recovery via Deep Interpreted RF Sensing. In *Proc. of the 27th ACM MobiCom*. 392–405.

[11] Chi-Wai Chow, Yang Liu, Chien-Hung Yeh, Yun-Han Chang, Yun-Shen Lin, Ke-Ling Hsu, Xin-Lan Liao, and Kun-Hsien Lin. 2021. Display Light Panel and Rolling Shutter Image Sensor Based Optical Camera Communication (OCC) Using Frame-Averaging Background Removal and Neural Network. *OSA/IEEE Journal of Lightwave Technology* 39, 13 (2021), 4360–4366.

[12] Yuanhao Cui, Fan Liu, Xiaojun Jing, and Junsheng Mu. 2021. Integrating Sensing and Communications for Ubiquitous IoT: Applications, Trends, and Challenges. *IEEE Network* 35, 5 (2021), 158–167.

[13] Christos Danakis, Mostafa Afgani, Gordon Povey, Ian Underwood, and Harald Haas. 2012. Using a CMOS Camera Sensor for Visible Light Communication. In *Proc. of IEEE GLOBECOM Workshops*. 1244–1248.

[14] CORE-Lens Dataset. 2022. https://github.com/XavierLiu888/Dataset-V1.

[15] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. 2020. RF-Net: A Unified Meta-Learning Framework for RF-Enabled One-Shot Human Activity Recognition. In *Proc. of the 18th ACM SenSys*. 517–530.

[16] GigaDevice. 2022. GD32F330G8U6 - GD32 ARM Cortex-M4 Microcontroller. https://www.gigadevice.com/microcontroller/gd32f330g8u6/. Online; accessed 4 March 2022.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proc. of NIPS*. 2672–2680.

[18] Jie Hao, Yanbing Yang, and Jun Luo. 2016. CeilingCast: Energy Efficient and Location-Bound Broadcast through LED-Camera Communication. In *Proc. of the 35th IEEE INFOCOM*. 1–9.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. of the 29th IEEE/CVF CVPR*. 770–778.

[20] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *Proc. of the 5th ICLR*. 1–22.

[21] Ke-Ling Hsu, Yu-Chun Wu, Yu-Cheng Chuang, Chi-Wai Chow, Yang Liu, Xin-Lan Liao, Kun-Hsien Lin, and Yi-Yuan Chen. 2020. CMOS Camera Based Visible Light Communication (VLC) using Grayscale Value Distribution and Machine Learning Algorithm. *Opt. Express* 28, 2 (2020), 2427–2432.

[22] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *Proc. of the 30th IEEE/CVF CVPR*. 4700–4708.

[23] Huawei Device Co., Ltd. 2022. Huawei Mate 30 Pro. https://consumer.huawei.com/sg/phones/mate30-pro/. Online; accessed 4 March 2022.

[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Proc. of the IEEE CVPR*. 1125–1134.

[25] Shuang Jiang, Zhiyao Ma, Xiao Zeng, Chenren Xu, Mi Zhang, Chen Zhang, and Yunxin Liu. 2020. SCYLLA: QoE-aware Continuous Mobile Vision with FPGA-based Dynamic Deep Neural Network Reconfiguration. In *Proc. of the 39th IEEE INFOCOM*. 1369–1378.

[26] Cristo Jurado-Verdu, Victor Guerra, Vicente Matus, Jose Rabadan, and Rafael Perez-Jimenez. 2021. Convolutional autoencoder for exposure effects equalization and noise mitigation in optical camera communication. *Opt. Express* 29, 15 (Jul 2021), 22973–22991.

[27] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. In *Proc. of ICLR*. 1–14.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. Imagenet Classification with Deep Convolutional Neural Networks. *Commun. ACM* 60, 6 (2017), 84–90.

[29] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. 2018. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. In *Proc. of the 6th ICLR*. 1–16.

[30] Siddharth Krishna Kumar. 2017. On Weight Initialization in Deep Neural Networks. *arXiv preprint arXiv:1704.08863* (2017).

[31] Hui-Yu Lee, Hao-Min Lin, Yu-Lin Wei, Hsin-I Wu, Hsin-Mu Tsai, and Kate Ching-Ju Lin. 2015. Rollinglight: Enabling Line-of-Sight Light-to-Camera Communications. In *Proc. of the 13th ACM MobiSys*. 167–180.

[32] Yun-Shen Lin, Yang Liu, Chi-Wai Chow, Yun-Han Chang, Dong-Chang Lin, Shao-Hua Song, Ke-Ling Hsu, and Chien-Hung Yeh. 2021. Z-Score Averaging Neural Network and Background Content Removal for High Performance Rolling Shutter based Optical Camera Communication (OCC). In *Optical Fiber Communication Conference (OFC) 2021*. F1A.4.

[33] Hongbo Liu, Bo Liu, Cong Shi, and Yingying Chen. 2017. Secret Key Distribution Leveraging Color Shift Over Visible Light Channel. In *Proc. of the IEEE CNS*. 1–9.

[34] Liqiong Liu and Lian-Kuan Chen. 2021. Li-poster: Real-time Non-line-of-sight Optical Camera Communication for Hand-held Smartphone Applications. In *Proc. of OSA OFC*. M1B.9.

[35] Liqiong Liu, Yang Hong, and Lian-Kuan Chen. 2018. A Frame Averaging based Signal Tracing (FAST) Algorithm for Optical Camera Communications. *Asia Communications and Photonics Conference (ACP) 2018*, Su3D.3.

[36] Xiaochen Liu, Pradipta Ghosh, Oytun Ulutan, BS Manjunath, Kevin Chan, and Ramesh Govindan. 2019. Caesar: Cross-Camera Complex Activity Recognition. In *Proc. of the 17th ACM SenSys*. 232–244.

[37] Ziwei Liu, Lin Yang, Yanbing Yang, Rengmao Wu, Lei Zhang, Liangyin Chen, Die Wu, and Jun She. 2021. Improved Optical Camera Communication Systems using a Freeform Lens. *Opt. Express* 29, 21 (2021), 34066–34076.

[38] Minghuang Ma, Haoqi Fan, and Kris M Kitani. 2016. Going Deeper into First-Person Activity Recognition. In *Proc. of the 29th IEEE CVPR*. 1894–1903.

[39] Mazda Moayeri, Phillip Pope, Yogesh Balaji, and Soheil Feizi. 2022. A Comprehensive Study of Image Classification Model Sensitivity to Foregrounds, Backgrounds, and Visual Attributes. In *Proc. of the IEEE CVPR*. 19087–19097.

[40] Huynh Nguyen, Archan Misra, and Youngki Lee. 2016. LightSense: Exploiting Smart Bulbs for Practical Multimodal Localization. In *Proc. of the 14th IEEE PerCom*. 1–4.

[41] Jingyi Ning, Lei Xie, Yi Li, Yingying Chen, Yanling Bu, Baoliu Ye, and Sanglu Lu. 2022. MoiréPose: Ultra High Precision Camera-to-Screen Pose Estimation based on Moiré Pattern. In *Proc. of the 28th ACM MobiCom*. 1–14.

[42] NVIDIA. 2022. GET SUPER POWERS GEFORCE RTX 2070 SUPER. https://www.nvidia.com/en-me/geforce/graphics-cards/rtx-2070-super/. Online; accessed 4 March 2022.

[43] Misa Ogura and Ravi Jain. 2022. FlashTorch. https://github.com/MisaOgura/flashtorch. Online; accessed 4 March 2022.

[44] A Pai, A Veeraraghavan, and A. Sabharwal. 2021. HRVCam: Robust Camera-based Measurement of Heart Rate Variability. *J. Biomed Opt* 26 (2021), 1–23.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv preprint arXiv:1912.01703* (2019).

[46] PyTorch. 2022. PyTorch Mobile. https://pytorch.org/mobile/home/. Online; accessed 4 March 2022.

[47] Nasir Saeed, Shuaishuai Guo, Ki-Hong Park, Tareq Y. Al-Naffouri, and Mohamed-Slim Alouini. 2019. Optical Camera Communications: Survey, Use Cases, Challenges, and Future Trends. *Physical Communication* 37 (2019), 100900.

[48] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv preprint arXiv:1312.6034* (2013).

[49] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).

[50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *Proc. of IEEE CVPR*. 1–9.

[51] Zhao Tian, Charles J. Carver, Qijia Shao, Monika Roznere, Alberto Quattrini Li, and Xia Zhou. 2020. PolarTag: Invisible Data with Light Polarization. In *Proc. of the 21st HotMobile*. 74–79.

[52] UNW. 2022. UMW SI2310A N-Channel Power MOSFET. https://www.semiee.com/file/Source10/UMW-SI2310A.pdf. Online; accessed 4 March 2022.

[53] Tim Van Erven and Peter Harremos. 2014. Rényi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory* 60, 7 (2014), 3797–3820.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. *Proc. of NIPS* 30 (2017), 1–11.

[55] Purui Wang, Lilei Feng, Guojun Chen, Chenren Xu, Yue Wu, Kenuo Xu, Guobin Shen, Kuntai Du, Gang Huang, and Xuanzhe Liu. 2020. Renovating Road Signs for Infrastructure-to-Vehicle Networking: A Visible Light Backscatter Communication and Networking Approach. In *Proc. of the 26th ACM MobiCom*. 1–13.

[56] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y. Zhao. 2021. Backdoor Attacks Against Deep Learning Systems in the Physical World. In *Proc. of the IEEE CVPR*. 6202–6211.

[57] Yue Wu, Purui Wang, Kenuo Xu, Lilei Feng, and Chenren Xu. 2020. Turboboosting Visible Light Backscatter Communication. In *Proc. of the ACM SIGCOMM*. 186–197.

[58] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2021. Noise or Signal: The Role of Image Backgrounds in Object Recognition. In *Proc. of the 9th ICLR*. 1–28.

[59] Chenren Xu, Shuang Jiang, Guojie Luo, Guangyu Sun, Ning An, Gang Huang, and Xuanzhe Liu. 2022. The Case for FPGA-Based Edge Computing. *IEEE Transactions on Mobile Computing* 21, 7 (2022), 2610–2619.

[60] Xieyang Xu, Yang Shen, Junrui Yang, Chenren Xu, Guobin Shen, Guojun Chen, and Yunzhe Ni. 2017. PassiveVLC: Enabling Practical Visible Light Backscatter Communication for Battery-Free IoT Applications. In *Proc. of the 23rd ACM MobiCom*. 180–192.

[61] Yanbing Yang, Jie Hao, and Jun Luo. 2017. CeilingTalk: Lightweight Indoor Broadcast Through LED-Camera Communication. *IEEE Trans. on Mobile Computing* 16, 12 (2017), 3308–3319.

[62] Yanbing Yang, Jie Hao, Jun Luo, and Sinno Jialin Pan. 2017. CeilingSee: Device-free occupancy inference through lighting infrastructure based LED sensing. In *Proc. of the 15th IEEE PerCom*. 247–256.

[63] Yanbing Yang and Jun Luo. 2018. Boosting the Throughput of LED-Camera VLC via Composite Light Emission. In *Proc. of the 37th IEEE INFOCOM*. 315–323.

[64] Yanbing Yang, Jun Luo, Chen Chen, Zequn Chen, Wen-De Zhong, and Liangyin Chen. 2021. Pushing the Data Rate of Practical VLC via Combinatorial Light Emission. *IEEE Trans. on Mobile Computing* 20, 5 (2021), 1979–1992.

[65] Yanbing Yang, Jiangtian Nie, and Jun Luo. 2017. ReflexCode: Coding with Superposed Reflection Light for LED-Camera Communication. In *Proc. of the 23rd ACM MobiCom*. 193–205.

[66] Zitong Yu, Wei Peng, Xiaobai Li, Xiaopeng Hong, and Guoying Zhao. 2019. Remote Heart Rate Measurement from Highly Compressed Facial Videos: an End-to-End Deep Learning Solution with Video Enhancement. In *Proc. of the IEEE/CVF ICCV*. 151–160.

[67] Tianyue Zheng, Zhe Chen, Shujie Zhang, Chao Cai, and Jun Luo. 2021. MoRe-Fi: Motion-robust and Fine-grained Respiration Monitoring via Deep-Learning UWB Radar. In *Proc. of the 19th ACM SenSys*. 111–124.

[68] Shilin Zhu, Chi Zhang, and Xinyu Zhang. 2017. Automating Visual Privacy Protection Using a Smart LED. In *Proc. of the 23rd ACM MobiCom*. 329–342.