Face Detection and Verification Using Lensless Cameras

Jasper Tan, Li Niu, Jesse K. Adams, Vivek Boominathan, Jacob T. Robinson, Richard G. Baraniuk, and Ashok Veeraraghavan

***Abstract*—Camera-based face detection and verification have advanced to the point where they are ready to be integrated into myriad applications, from household appliances to Internet of Things (IoT) devices to drones. Many of these applications impose stringent constraints on the form-factor, weight, and cost of the camera package that cannot be met by current-generation lens-based imagers. Lensless imaging systems provide an increasingly promising alternative that radically changes the form-factor and reduces the weight and cost of a camera system. However, lensless imagers currently cannot offer the same image resolution and clarity of their lens-based counterparts. This paper details a first-of-its-kind evaluation of the potential and efficacy of lensless imaging systems for face detection and verification. We propose the usage of existing deep learning techniques for face detection and verification that account for the resolution, noise, and artifacts inherent in today's lensless cameras. We demonstrate that both face detection and verification can be performed with high accuracy from the images acquired from lensless cameras, which paves the way to their integration into new applications. A key component of our study is a dataset of 24,112 lensless camera images captured using FlatCam of 88 subjects in a range of different operating conditions.**

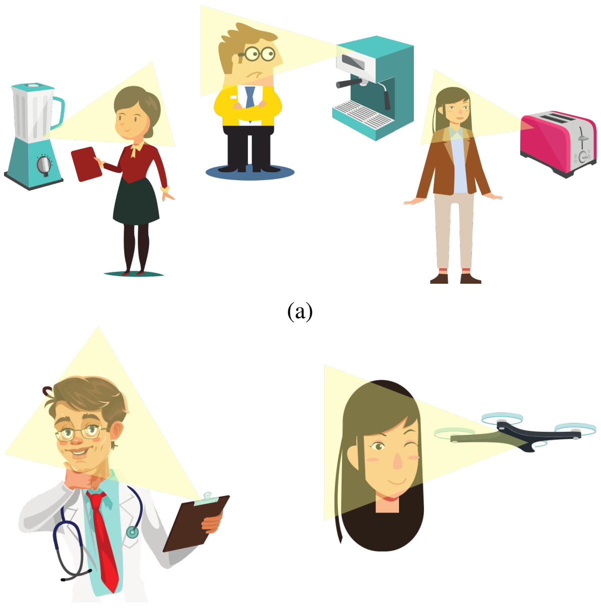
***Index Terms*—face detection, face verification, lensless imaging, coded aperture, machine vision, deep learning.**

I. Introduction

The last few decades have seen an explosion in the use of cameras and other imaging devices. It is estimated that in the year 2017 alone more than a billion camera modules were sold, most of them integrated into mobile systems such as phones, tablets, and other devices. The miniaturization and rapidly decreasing cost of camera modules have been the primary drivers of this scaling. With the rise in use and integration of cameras, their role in our lives has also changed significantly. Most cameras no longer take photographs, but instead are used as sensors to provide inferential inputs for a diverse range of applications, from biometrics (face recognition, authentication) to surveillance and security.

While modern camera modules can be thin ($\approx$ 5mm) and inexpensive ($\approx$ \$15)[1], many emerging applications, such as the Internet of Things (IoT), surveillance, disaster recovery using drones, etc., impose stringent constraints on size, weight, and cost that cannot be met with today's lens-based imaging systems. Consider the integration of cameras into household electronics and electric appliances, such as thermostats, coffee machines, toasters, refrigerators, etc. Given the cost-constraints in these devices, integration is only feasible if we can realize an order of magnitude reduction in the cost of camera modules. Consider also the integration of cameras into drones for distributed sensing in applications such as disaster recovery or in very thin objects such as credit cards. Given the weight and volume constraints in these devices, integration is only possible if we can realize an order of magnitude reduction in the weight and size of camera modules.

Fortunately, remarkable progress has been made in the past five years on *lensless imaging*, resulting in imaging devices that are an order of magnitude thinner ($\approx$ 1mm) and less


(a)

(b)

Fig. 1: A lensless camera's radically reduced form-factor and size/weight profile can enable face detection and verification in emerging applications. (a) Internet of Things (IoT) applications can distribute a large number of inexpensive sensors throughout an environment. (b) Drones, UAVs and other mobile platforms can more easily satisfy their stringent geometric and weight constraints. (Graphics by Freepik.)

J. Tan, J. K. Adams, V. Boominathan, J. T. Robinson, R. G. Baraniuk, A. Veeraraghavan are with the department of Electrical and Computer Engineering at Rice University, Houston, TX, 77005 (e-mail: jaspertan@rice.edu, jkadams@rice.edu, vivekb@rice.edu, jtrobinson@rice.edu, richb@rice.edu, vashok@rice.edu).

L. Niu is with the Department of Computer Science and Technology at Shanghai Jiao Tong University, Shanghai, China.

This work was supported in part by DARPA under grant HR0011-17-C-0026 and by NSF under grants CCF-1527501 and IIS-1652633.

[1] Estimates are based on breakdown analyses of the camera modules in current mobile phones [1].

expensive at-scale (< $1) [2]–[5]. The main technological advantage of the emerging class of lensless imaging devices is that, by eliminating the need for a lens, they can reduce the weight and thickness of the device by an order of magnitude. Moreover, since they can be completely fabricated using traditional semiconductor fabrication processes, they can reduce the cost by an order of magnitude to a lens-based camera.

Without a lens, focusing in a lensless camera is performed through an algorithm that reconstructs an image from coded measurements. These algorithms are invariably imperfect and result in images with lower spatial resolution and reconstruction-algorithm-dependent artifacts. Additional degradation includes potentially reduced signal-to-noise ratio (SNR), limited dynamic range, and blurring.

An important research question that needs to be addressed is thus whether the quality and resolution of lensless camera images are sufficient for their adoption in the emerging applications discussed above. In this paper, we delve into this question deeply in the context of face detection and verification. We focus on these tasks because of their growing maturity and application in real-world systems. Our goal is to explore the feasibility of performing face detection and verification with lensless cameras by building and studying an imaging system that couples a lensless camera with existing deep learning inference techniques. Since the performance of deep learning depends strongly on the amount of training data in order to identify the patterns induced by faces while remaining unaffected by artifacts and noise, we present two techniques to convert any standard training dataset into a lensless camera image database: physically capturing images displayed on a screen and digitally simulating them. Both methods capture the characteristic noise sources, resolutions, and artifacts of lensless cameras that differ significantly from those of conventional cameras.

To evaluate the performance of our system in realistic scenarios, we build a real face dataset, the *FlatCam Face Dataset (FCFD)*, with 88 subjects and 24,112 images acquired using the FlatCam lensless system [2], [6] including multiple lighting conditions, expressions, angles, capture times, and backgrounds. Despite the lower image quality of our system, we show that we can still achieve reasonable results for the tasks of face detection and face verification.

The main technical contributions of this paper are as follows:

1) We perform a first-of-its-kind study to evaluate the performance of lensless imaging systems in face detection and verification.
2) We demonstrate techniques (simulation and display-captured) for generating training datasets for inference algorithms (such as deep learning based face analysis algorithms) and study the effect of their approximations on inference accuracy.
3) We demonstrate how with proper training images, existing deep learning methods can achieve reasonable results for face detection and face verification on images captured by a lensless imaging system.
4) We experimentally evaluate the performance difference for face detection and face verification between one particular lensless imaging system (FlatCam) and a conventional camera.
5) We acquire and study a first-of-its-kind lensless camera face dataset (that will be made publicly available) that contains various lighting conditions, facial expressions, angles, backgrounds, and resolutions.

II. Related Work

A. Face Detection and Verification

The goal of face detection is to infer whether a given image contains faces and if so to provide their locations using, for example, bounding boxes. Following the impactful algorithm introduced by [7], many methods have been developed to solve the face detection problem, as described in the survey papers of [8], [9]. Recently, deep learning methods have come to the fore; they offer state-of-the-art performance for face detection [10]–[14] using convolutional neural networks to perform tasks such as classifying face vs. non-face regions and predicting bounding box coordinates. In this work, we follow the method of [11] and utilize Faster R-CNN [15], a deep learning framework that has been successful for general object detection, for face detection.

The goal of face verification is to identify whether two given face images contain the same identity or not. Some of the earlier efforts are described by [16]–[18]. As with face detection, deep learning techniques offer state-of-the-art performance for face recognition and verification [19]–[24]. Briefly, the idea is to use a convolutional neural network to extract features from both images that are then input into a classifier, which can be as simple as calculating the distance between the two sets of features.

The above works focus on face detection and verification from imagery captured by a conventional lens-based camera. Another line of work has used data from other imaging devices [25]. For example, some studies have aimed to perform face recognition using thermal infrared imagery [26], [27], with the argument that thermal sensors are less sensitive to ambient light variations than visible light cameras. Researchers have also tried improving face detection and verification results using depth sensors [28]. While these studies replace traditional cameras with (often more expensive) detectors/set ups to improve accuracy, we use lensless technology to achieve face detection and verification with cheaper and thinner imaging devices.

B. Lensless Imaging

While lensless imaging has been widely used for imaging non-visible wavelengths [29], [30], we focus here on lensless visible light imaging. A lensless imaging system captures measurements using either a bare imaging sensor or a sensor plus a non-lensing mask that modulates the light such that recovery of the desired image is possible. Examples of masks are compressive samplers [31], [32], programmable LCDs [33], phase gratings [4], spatial light modulators [34], [35], and diffusers [36]. In the works of [2], [6], several of the authors built a new lensless imaging device called *FlatCam* by placing

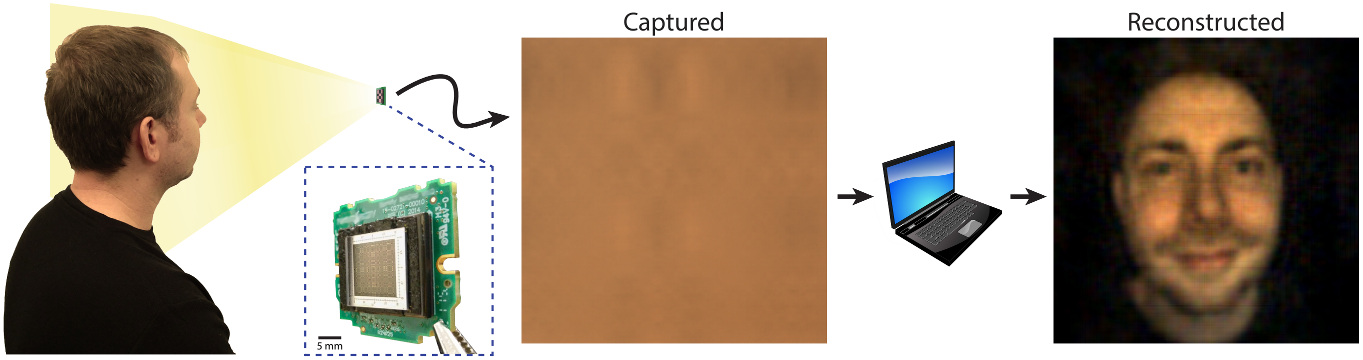


Fig. 2: The FlatCam lensless imager places a thin mask of apertures atop a bare image sensor. Light enters through the multiple apertures, and the sensor captures a superposition of shifted mask shadows. A reconstruction algorithm unveils the desired scene image.

a binary amplitude mask containing a pattern of opaque and transparent features less than 1.5mm from the imaging sensor.

We use a FlatCam prototype for this work, since it has been demonstrated to achieve a very thin form-factor in a simple and inexpensive design. Such features allow its use in applications with strict size and cost constraints wherein lens-based cameras may be infeasible. We note that our experiments and results in this paper are specific to the FlatCam and may not accurately characterize face detection and verification with other lensless imaging systems that operate differently.

There has been limited previous work in inference from a lensless camera. The work in [37] classified images of digits displayed on a screen that were captured by a bare image sensor. In [38], the location of a vertical bar was estimated using a lensless imaging system with a mask containing a phase grating. To the best of our knowledge, our work is the first to perform face detection and verification with a lensless camera and also the first to perform an inference task on a real captured dataset of natural images.

Reconstructing lensless images is an inverse problem in imaging, and there have been works that propose deep learning methods for solving such inverse problems [39]–[45]. In our work, we do not use deep learning methods to perform the image reconstruction but use them on the reconstructed images themselves. We do this for multiple reasons. First, the method we use to reconstruct FlatCam images is much more computationally simple. Second, FlatCam devices with different masks can yield very different measurements for the same scene, but their reconstructed images are quite similar. Thus, performing inference directly on the measurements may require training on each device individually, but performing inference on the reconstructed images only requires training on reconstructed images provided by one device (as verified in the Supplementary Material).

III. Lensless Imaging with FlatCam

The key idea we use to reduce the size and cost of an imaging system is to exclude the lens completely. In this work, we used a prototype of *FlatCam*, which was introduced in [2], [6]. In this section, we provide a brief description on how FlatCam produces images. For more detailed explanation, see the above papers. In the remainder of this paper, we will refer to images captured with FlatCam as "lensless images" and images captured with a conventional lens-based camera as "standard images".

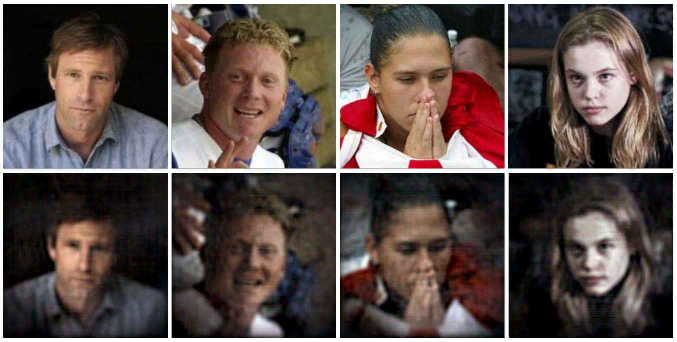


Fig. 3: Images from the LFW dataset (top) were displayed on a screen and then captured and reconstructed (bottom) by FlatCam. The lensless reconstructions feature artifacts and non-idealities that are not present in standard camera images.

Our FlatCam prototype consists of a Point Grey Flea3 camera with 1.3 MP e2v EV76C560 CMOS sensor with a pixel size of $5.3\,\mu\text{m}$. We placed an amplitude mask ≈ 1.2mm above the sensor using chrome (opaque features) on glass (apertures). The pattern of the amplitude mask is a crop of a modified uniformly redundant array (MURA) [46] of length 1277, with smallest apertures of $20\,\mu\text{m} \times 20\,\mu\text{m}$.

To understand how FlatCam works, consider imaging a point source, like an LED. Light from this point source enters the image sensor only through the apertures on the mask, and thus the system's point spread function (PSF) is a shadow of the mask pattern. Capturing a more complex scene then results in a sensor measurement that is a superposition of shifted mask patterns of different intensities. If the mask pattern is the outer product of two vectors, then FlatCam can be approximately modeled using a separable linear transform [2], [35]:

$$Y = \Phi_L X \Phi_R^T + N, \tag{1}$$

where the matrix Y represents the two-dimensional (2D) sensor measurements, the matrix X represents the scene

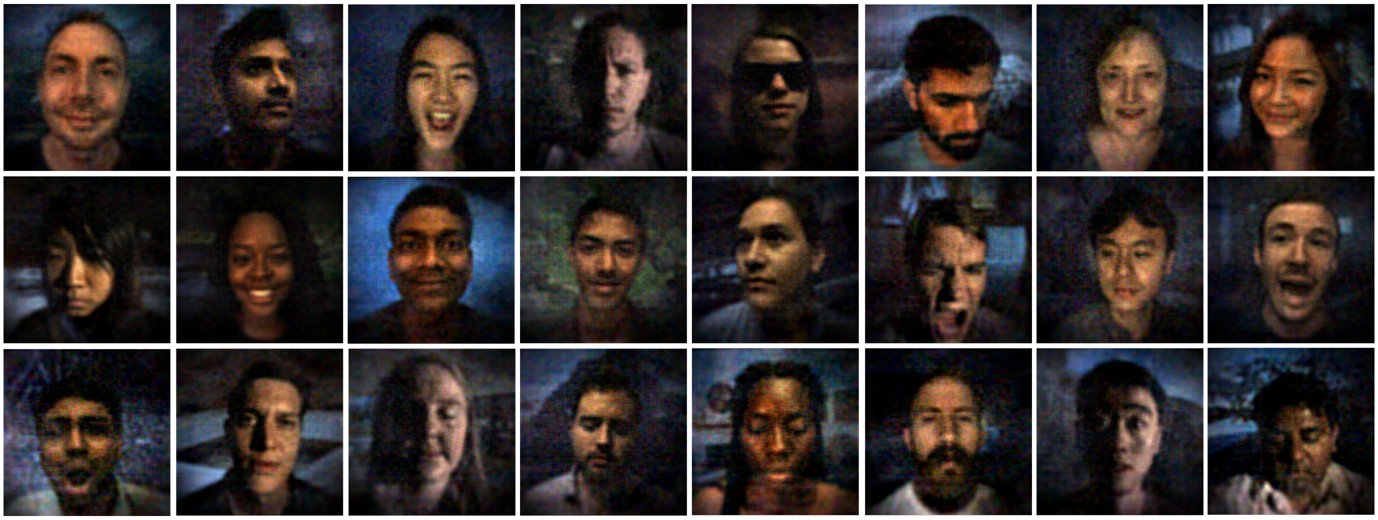


Fig. 4: Sample images of different subjects from the FlatCam Face Dataset (FCFD). The images contain variations in expression, lighting, angle, scale, and background.

irradiance from the plane of interest, N represents additive noise, and Φ_L and Φ_R encode the PSFs that form the linear model. We refer to this as the "FlatCam forward model". In our design, we use a 500×620 portion of the sensor (i.e. dimensions of Y) for each color channel and we calibrate to reconstruct a scene X of size 256×256. We obtain Φ_L and Φ_R using the calibration scheme described by [2].

Given the measurements Y, we reconstruct the scene X by solving the following regularized least squares problem:

$$X = \arg\min_X ||Y - \Phi_L X \Phi_R^T||_F^2 + \lambda ||X||_F^2 \qquad (2)$$

for each color channel individually. Here $\|\cdot\|_F$ denotes the Frobenius norm. The regularization enables the reconstruction algorithm to function even when Φ_L and Φ_R are not well conditioned. [2] derive the closed form solution of this problem:

$$\hat{X} = V_L[(\Sigma_L^T U_L^T Y U_R \Sigma_R)./(\sigma_L \sigma_R^T + \lambda \mathbf{1}\mathbf{1}^T)]V_R^T, \qquad (3)$$

where $\Phi_L = U_L \Sigma_L V_L^T$ and $\Phi_R = U_R \Sigma_R V_R^T$ are the singular value decompositions of Φ_L and Φ_R, respectively, σ_L and σ_R are vectors containing the values on the diagonals of Σ_L^2 and Σ_R^2, and where ./ denotes elementwise division.

In this paper, we use $\lambda = 3 \times 10^{-4}$ for all color channels based on the visual quality of a few reconstructed images. Fig. 2 details the operation pipeline of FlatCam, as well as an example of a FlatCam measurement and its corresponding reconstruction. Samples of reconstructed FlatCam images of a screen projecting face images are shown in Fig. 3.

FlatCam's reconstruction procedure involves a few small matrix multiplications, and thus it is computationally inexpensive. Although the scene images could be reconstructed by employing more sophisticated optimization techniques, the images produced by the above method have a sufficient level of quality for reasonable face detection and verification accuracy.

Compared with conventional lens-based cameras, imaging with lensless cameras has several drawbacks. First, the transmission of light may not exactly follow the separable linear model due to factors such as diffraction, resulting in reconstruction errors. Second, since light from an object in the scene reaches multiple (if not all) pixels on the sensor, the dynamic range will be lower than that of a conventional camera; in particular, one bright object can saturate the entire sensor instead of just a small number of pixels. Third, reconstructed images will lower resolution due to the very small distance between the mask and the sensor, which causes small movements of a point source to yield very similar measurements. It is therefore an open question as to whether thin lensless imaging systems are up to the tasks of face detection and verification.

IV. FlatCam Face Dataset

For an inference task, a large enough test dataset is required for a meaningful evaluation. There exists no such dataset of lensless camera images. Therefore, we acquired the first (to the best of our knowledge) real face dataset taken using a lensless imaging device. The *FlatCam Face Dataset (FCFD)* contains 24,112 images of 88 subjects (274 images per subject), which we use to evaluate face verification performance.

Generally speaking, there are two types of face verification datasets: "controlled" datasets and "in the wild" datasets. In both, it is important to have multiple variations in the images such as different lighting conditions, angles, facial expressions, etc. Such diversity is needed to test algorithms and methods that will be deployed in real-life settings where such variations naturally occur.

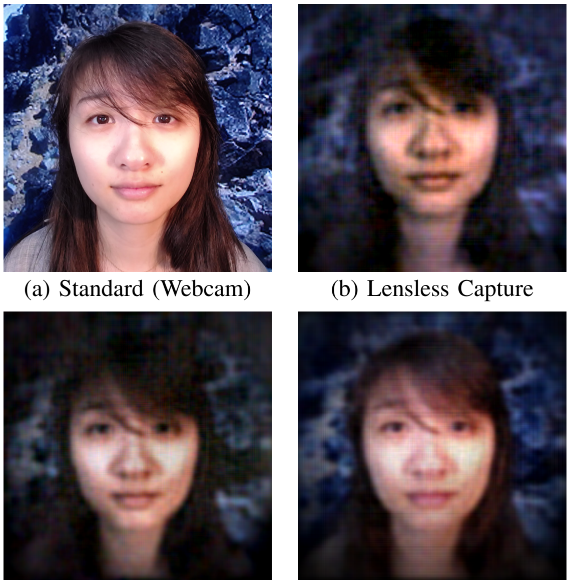
In a "controlled" dataset, subjects come to a lab or studio to have their images taken. Since this can seem like an artificial setting, researchers deliberately add variations by having different types of illumination devices, asking the subjects to make different facial expressions, etc. Examples of such datasets are listed in Table I (note: this list is not exhaustive). In an "in the wild" dataset, images are obtained from public sources, such as the world wide web. Such photos are generally captured in natural settings and hence already contain natural variations such as different backgrounds, different lighting

TABLE I: Commonly used controlled face datasets and the intra-class variations they contain. “Expressions” refers to the number of different facial expressions per subject, “Lighting” refers to the number of lighting conditions used, “Angles” refers to different viewpoints of the face, “Time between sessions” refers to the amount of time in between different capture sessions for the same subject, and “Occlusions” refers to the use of scarves, sunglasses, or other occluding accessories. Our dataset contains most of the intra-class variations included in other datasets.

Dataset	Subjects	Total images	Expressions	Lighting	Angles	Time between sessions	Occlusions
FERET [47]	1199	14126	2	2	5	1 year	None
AR Face Database [48]	126	4026	4	4	1	2 weeks	Sunglasses, scarf
FRGC Data Set [49]	466	51433	4	2	1	1 year	None
ORL Database	40	400	Up to 10	Up to 10	10	Unspecified	None
Yale Face [50]	15	165	6	3	1	1 session only	Glasses
Yale Face B [51]	10	5760	1	65	9	1 session only	None
CMU Multi-PIE [52]	68	41368	6	19	15	Unspecified	None
FCFD (this paper)	88	24112	9	10	11	2 weeks	Sunglasses

conditions, and different facial expressions. Examples of these datasets are the Labeled Faces in the Wild (LFW) dataset [53], [54], the VGG face database [19], CASIA-WebFace [55], and the megaface benchmark [56].

Since there are few FlatCam face images publicly available on the web, the FCFD is, by necessity, a controlled dataset. As such, we incorporated multiple variations in the face images. To build the FCFD, we captured images of subjects sitting 23—38cm from the FlatCam while using 10 different lighting conditions. For each lighting condition, we captured 10 expressions: neutral, smiling, angry, screaming, closed eyes, sad, sleepy, surprised, winking, and wearing sunglasses (for occlusion). For each lighting condition, we also captured 8 angles; the participant looks at 8 uniformly spread locations on a circle such that the angle their direction makes with the camera is approximately 35°. For each of the lighting conditions, neutral and smiling images were also captured for when the participant is closer (approximately 15cm) to the camera for scale variation. At least two weeks after the first session, participants had their images captured again with 6 of the lighting conditions, 4 of the angles, and 5 of the expressions. More details on the variations can be found in the supplementary material. In each image, the subject is sitting in front of a television screen displaying a random faceless background image from the ImageNet database. While backgrounds in real scenarios are not planar like a television screen, it has been shown that for large enough distances (> 30cm), differences in scene depths yield small differences for FlatCam measurements [57]. We compare the variations we included in our dataset with those contained in other popular controlled face datasets in Table I. Sample images from the FCFD can be found in Fig. 4. Each image in the dataset was also captured with a Logitech C930e webcam for a lens-based comparison to the lensless images.



(a) Standard (Webcam)

(b) Lensless Capture

(c) Display-captured

(d) Simulated

Fig. 5: The three different types of training images we explored in this work compared to a real FlatCam capture. (a) A subject captured with a webcam. (b) The same subject captured with an actual FlatCam. (c) FlatCam capture of a screen projecting the webcam image. (d) Simulated lensless camera image obtained by applying Eq. 1 to the webcam image and reconstructing with Eq. 3.

V. Obtaining Lensless Training Images

While the comprehensive FCFD is key to testing our face detection/verification system, the training process of the system’s deep networks needs to be fueled by a large amount of training data. Many face datasets, such as the WIDER face dataset [58], the Annotated Facial Landmarks in the Wild (AFLW) dataset [59], and the VGG face dataset [19], are available for training algorithms for face detection and verification. They contain a large number of images and include many natural variations. However, these images are captured by lens-based cameras and do not reflect the unique characteristics of lensless images. It would take a long time to physically capture and annotate an equivalent lensless face training dataset (for example, the VGG face dataset contains more than 2 million images), and thus, it would be more convenient to quickly obtain a lensless version of existing standard face datasets.

In this work, we propose a method of converting a standard (lens-based) face training dataset into a lensless one by displaying the images on a monitor and capturing them using a FlatCam prototype. The images are real in that they come from a physical lensless camera, but they are still simply projected images rather than images of real physical faces. We scale the standard images on the monitor such that the larger dimension fills the field of view of FlatCam while keeping the same aspect ratio. Thus, the bounding box information for the training images can be preserved by scaling to the dimension of the reconstructed FlatCam images. We refer to a dataset generated this way as "display-captured". Indeed display-captured images have a mismatch compared to true lensless images of real human faces, but our experiments show that using these training data can achieve reasonable results for face detection and verification on real lensless images.

Some disadvantages of the display-captured method are that it is time-consuming and that it requires a physical setup, and so we also explore digitally simulating the dataset entirely using the FlatCam forward model in Eq. 1. In particular, we transform a standard image X into $Y = \Phi_L X \Phi_R + N$, where Φ_L and Φ_R are obtained through the calibration scheme described by [2], and then we reconstruct using Eq. 3 with $\lambda = 8 \times 10^{-4}$ to obtain the simulated FlatCam image. We add zero-mean Gaussian random noise N such that $\frac{||\Phi_L X \Phi_R||}{||N||} = 10$, which we found closely models our true lensless images. Such a method is computationally simple and requires no real lensless images. However, its images will suffer from greater mismatches between the FlatCam forward model and reality. We refer to this approach to creating a dataset of lensless images as "simulated".

To visualize the images generated by the different methods, we captured an image of a subject with both a webcam and the FlatCam, as well as display-captured and simulated versions from the webcam capture (Fig. 5). The best training image would be one that most closely resembles the true FlatCam capture.

VI. Computational Architecture for Face Detection/Verification

After training data has been prepared, the next key ingredient is the computational architecture. To perform face detection and verification, we exploit recent methods from deep learning that have achieved excellent performance on standard test protocols. When choosing the deep learning techniques for face detection and verification, we focused on algorithms that are accessible and simple in both implementation and principle, so that our results are not exclusive to very specific techniques. We perform the two tasks separately and briefly describe the method we used for each task. Fig. 6 outlines the computational pipeline for our system.

A. Face Detection with Faster R-CNN

Apart from the detection networks specifically designed for faces, there has also been work on general object detection using deep learning such as YOLO [60], Faster R-CNN [15], and SSD [61]. A simple approach for face detection is to identify a method that has been successful for general object detection and apply it to detect faces. One recent method that achieves high accuracy for object detection is *Faster R-CNN*, which features an end-to-end convolutional neural network (CNN) method with efficient operation [15]. [11] performed face detection using Faster R-CNN and demonstrated very high accuracy. We deployed Faster R-CNN for lensless face detection based on code made available by [15] and [11].

Faster R-CNN comprises two main components: a Region Proposal Network (RPN) and a detection network. The RPN produces region proposals that are likely to contain objects of interest, while the detection network based on Fast R-CNN [62] classifies the proposals and regresses their bounding boxes to more precise locations and scales.

We train our Faster R-CNN model using the WIDER face dataset [58], which contains more than 12,000 training images, and the Annotated Facial Landmarks in the Wild (AFLW) dataset [59], which contains more than 25,000 training images of faces with multiple variations, as well as the bounding boxes for these faces. We train three different networks: one trained with the standard WIDER dataset (we found that including the AFLW dataset for standard images did not improve performance), one trained with display-captured lensless WIDER and AFLW datasets, and one trained with simulated lensless WIDER and AFLW datasets, allowing us to evaluate the performance difference of the various training datasets. Since our prototype FlatCam has a resolution (256 × 256 pixels) lower than many of the images in the WIDER dataset, for images containing very small faces, we use crops of the image instead of the entire image. We use the VGG16 network pretrained on the ImageNet dataset and the approximate joint training method described by [15]. The learning rate is set to 0.001 for the first 50k iterations and decreases to 0.0001 for 30k more iterations.[2]

B. Face Verification Through Classification CNN

One simple technique for face verification trains a CNN to perform a face inference task (such as identity classification), removes the final task-specific layers, and then treats the trained network as a feature extractor [19], [63], [64]. Since the network was trained for a face inference task, the extracted deep features should contain discriminative information on the original face images. After extracting deep features for two images, we predict whether they belong to the same identity or not by applying a threshold to a similarity measure calculated on the features. For simplicity, we use the negative ℓ_2 distance between two images' feature vectors as the similarity measure. This has been shown to achieve strong performance for face verification [19], [65].

We follow the method and architecture of [19] and use the 16-layer configuration-D CNN architecture from [66], which yields excellent results on the Imagenet challenge. For our training data, we use a subset of the cropped VGG face dataset introduced by [19], which contains approximately 900,000 out of the original 2.6M images of faces belonging to 2,622 identities. We initialize the CNN weights using a zero-mean random Gaussian distribution with a standard deviation of 10^{-2} and initialize the biases to zero. We perform the optimization using stochastic gradient descent with a batch size of 32 images and momentum coefficient of 0.9. We also apply batch normalization after every convolutional layer. We

[2] We used the implementation made available by [11]: https://github.com/playerkk/face-py-faster-rcnn

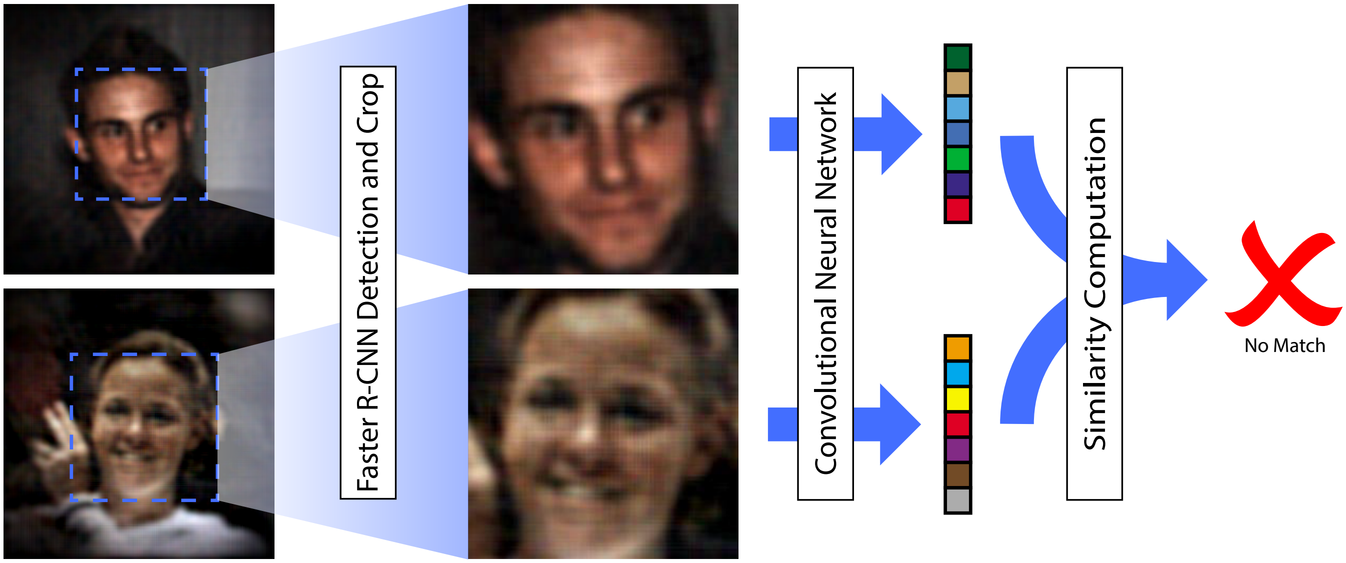


Fig. 6: Computational pipeline for our face detection and verification system. After capturing and reconstructing the face with FlatCam, face detection is performed using Faster-RCNN [15]. For face verification, we send the face region detected by Faster-RCNN to a convolutional neural network to yield a feature vector. To verify whether two faces have the same identity, we apply a threshold to the computed similarity between their two feature vectors.

set the initial learning rate to 10^{-2} and decrease it by a factor of 10 when the validation score is no longer decreasing, with the last learning rate being 10^{-4}. The training images are all of size 256×256, of which the center 224×224 cropped patches are sent through the network. We train the network to minimize the softmax loss for classification with the 2,622 classes using MatConvNet [67]. Similar to the method for face detection, we train three different verification networks using three different versions of the VGG training dataset (standard, display-captured, and simulated). After training, we remove the last classification layer such that given an image input, the CNN now outputs a 4096-dimensional feature vector.

To perform face verification given two images, we first preprocess the images by passing them through the Faster R-CNN face detector described above to obtain tight bounding boxes on the face regions. We then make these bounding boxes square by extending their shorter sides to match the longer sides, and we crop the resulting face patches out. The two cropped images are then resized to 256×256, and for each image, five 224×224 crops, one from each corner and one from the center patch, are passed through the CNN to obtain output vectors. The feature vector for an image is the average of its five output vectors, which we then normalize to have unit ℓ_2 norm. The similarity score of the two images is computed as the negative value of their feature vectors' ℓ_2 distance. If the similarity score is above a certain threshold, then the images are deemed to belong to the same identity. This method shares similarities with that proposed by [19].

VII. Experiments

We evaluate the performance of our lensless camera face detection and verification system on our captured FCFD dataset as well as on two common benchmark datasets used for face detection and face verification: the Face Detection Data Set and Benchmark (FDDB) [68] and Labeled Faces in the Wild (LFW) [53], [54] datasets.

TABLE II: Summary of FCFD experiments.

Experiment	**Property of interest**	**Comparisons made**
FE1	Most effective training data	Performance of networks trained by display-captured, simulated, and standard images
FE2	Effectiveness of face detection	Performance on uncropped images vs. images cropped via lensless face detection
FE3	Lens-based vs. lensless tradeoff	Performance on webcam images vs. FlatCam images
FE3	Robustness across different variations	Performance for different variations (including same-day vs. different-day images)
FE3	Particularly difficult variations	Performance on "Easy" set vs. "Complete" set.

A. Face Verification on FCFD

We perform three experiments on the FCFD to answer a number of questions regarding the performance of face verification on FlatCam images. A summary of these experiments are listed in Table II.

1) FCFD Testing Procedure: For each experiment, we build two different sets of images: the probe and gallery sets. We then perform face verification on all possible pairs of images between these two sets. The gallery set can be interpreted as saved template images for different subjects, and the probe set can be interpreted as the test images presented to the verification system in deployment. We perform both "Same Day" experiments and "Different Day" experiments, which indicate whether the images from the probe set were captured

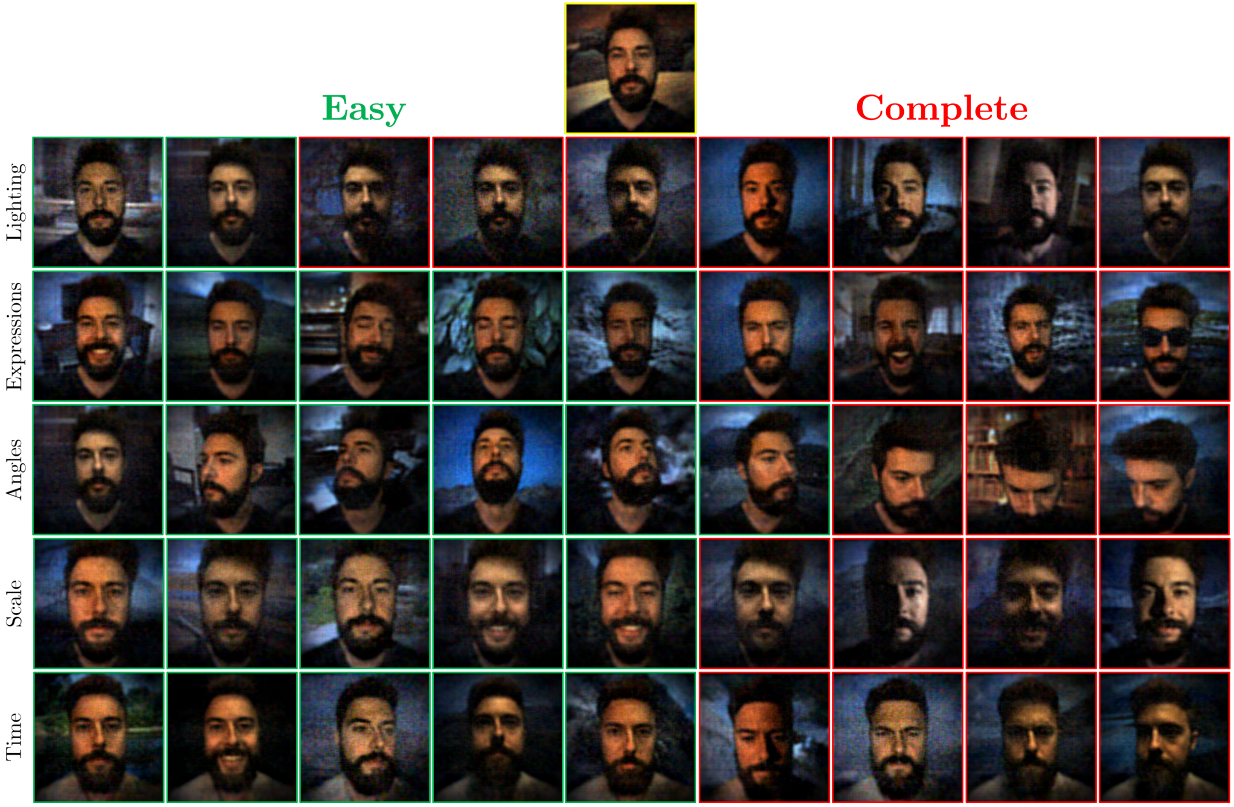


Fig. 7: Sample images of one subject from the FlatCam Face Dataset (FCFD). When testing on this dataset, we perform face verification by comparing one gallery image (top, yellow border) per subject with images from probe sets. From the dataset, we build two different probe sets: we use a subset containing simpler expressions, angles, and lighting ("Easy", green border), as well as the complete dataset ("Complete", red border). The probe sets include images acquired with variations in lighting, expressions, angles, and scale. It also contains images captured on a different day ("Time") from the gallery image.

on the same day or a different day (set apart by at least 2 weeks) from the gallery images. While the FCFD is ultimately designed for face verification, the images first pass through our display-captured Faster R-CNN face detector to crop the face region out for input to the verification algorithm. Thus, face detection is also an important component in this test.

The FCFD contains a number of natural variations, and we noticed in our experiments that some of these variations are more difficult to deal with than others in lensless face verification. Excluding these variations from the test can dramatically change the results. Generally, these variations are those where the subject is (a) looking down, (b) making a dramatic face expression (such as screaming or surprised), or (c) when there is no diffuse light source in front of the face. Given this, we generate two different test sets. The "Easy" test set excludes these tougher variations while the "Complete" test set contains all images of all variations. Examples of images for these variations for one subject are shown in Fig. 7. For our experiments, we build the gallery set by including one image per subject: a neutral expression facing the camera in a well-lit environment.

We report the results in two ways. The first way is by reporting the receiver operating characteristic (ROC) curve [69]. Particularly, by varying the threshold applied to the similarity values of the images' features, we obtain different pairs of True Positive Rates (TPR), the percentage of same-identity pairs correctly verified, and False Positive Rates (FPR), the percentage of different-identity pairs incorrectly verified as the same identity. The second way we report our results is by presenting the TPR given a fixed FPR. This is helpful for applications with stringent constraints on the FPR value.

2) *FE1: Effect of Training Data:* We first study the performance of the three different CNN models we have obtained using the three types of training data. For this, we use all images in the "Easy" set as the probe set. The results are reported in Table III and Fig. 8.

We observe that the best results are obtained by using display-captured images for the training set, followed by using simulated images, and finally by using standard images. This confirms the benefit of acquiring lensless training images for lensless face verification. The lower accuracy given by the simulated images compared to the display-captured images implies the model mismatch from Eq. 1, which is not very surprising, since such a model was built to have the reconstruction be computationally tractable rather than physically exact. The separable linear transformation may not accurately account for some intricacies of the physical lensless imaging process such as diffraction. However, simulating training data is still a method to achieve reasonable performance at low cost. All following FCFD experiments will use the CNN trained on

TABLE III: TPR with fixed FPR on the FCFD "Easy" dataset using the three CNN models trained with different types of training data.

Training Images	FPR=1%	FPR=0.1%
Standard	78.76%	47.45%
Display-captured	82.64%	55.25%
Simulated	80.66%	54.25%

TABLE IV: TPR with fixed FPR using the display-captured CNN on the FCFD "Easy" Dataset for both cropped and uncropped images.

Images	Same Day		Different Days	
	FPR=1%	FPR=0.1%	FPR=1%	FPR=0.1%
Cropped	82.64%	55.25%	73.94%	43.48%
Uncropped	62.32%	34.11%	47.46%	21.99%

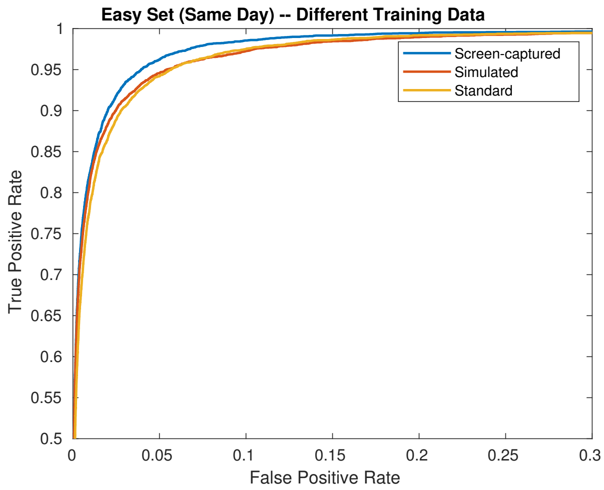


Fig. 8: ROC curves on the FCFD "Easy" dataset (Same Day) using the three CNN models trained with different types of training data. Display-captured training images yield the best results.

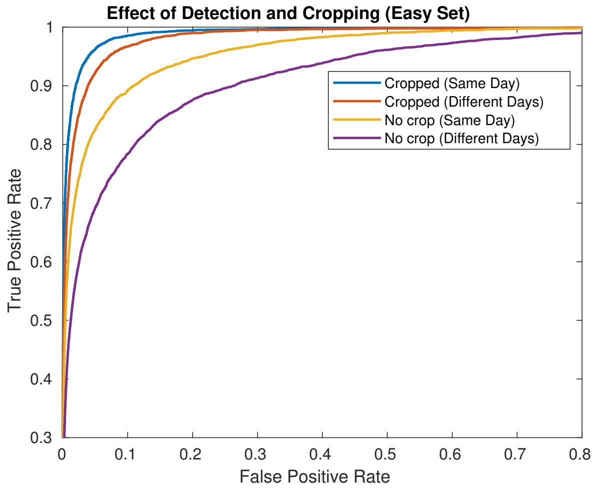


Fig. 9: ROC curves on FCFD test sets containing cropped face regions and original uncropped images. Cropped faces are located using the Faster R-CNN detector trained on display-captured images, which greatly improves accuracy and proves the effectiveness of our detector for detecting faces in lensless images.

display-captured images.

3) FE2: Effect of Face Detection: The first step to face verification is to run the images through our Faster R-CNN detector to crop the face regions out. In this experiment, we evaluate the effect of this step by running face verification on both the original (uncropped) FCFD images and on the cropped facial regions identified by our detector on the "Easy" test set. The results, presented in Fig. 9 and Table IV, show that cropping the face regions achieves a significant improvement in accuracy, which verifies the success of our detector in locating the faces in lensless images.

4) FE3: Variations: Next, we experiment in more detail with the different variations present in the FCFD. To perform the experiment for one variation, we build a probe set whose images differ from the gallery set by only that variation. For example, to perform the "Lighting" test, we choose a probe set containing all images where the subject is facing the camera with a neutral expression (i.e. same angles and expressions as the gallery) but with different lighting settings. The only exception to this is the "Scales" experiment, which is similar to our "Lighting" experiment except that the images in the probe set are captured when the subject is closer to the camera (≈ 15cm) than in the gallery set (23–38cm). Details on the variations included in each experiment are listed in the Supplementary Material. Table V lists the number of test images per subject for each experiment. Note that "All" also includes images that simultaneously have multiple variations (such as a different expression in a different lighting setting), which may not appear in the other experiments with only one variation. We also evaluate the results on the webcam (lens-based) captures of the FCFD using the CNN trained on standard images to allow comparison on lens-based vs. lensless face verification. Our results are shown in Fig. 10 and in Table VI.

The results first show that there is a performance decrease when using the lensless FlatCam compared to a lens-based webcam. This reveals the current tradeoff in lensless face verification: decreased performance for a much cheaper and thinner hardware system. However, the lensless face verification accuracies reported here may still be sufficient for many applications.

Next, the results show that the easiest variation the FlatCam handles is expression while the toughest is angle. The dramatic

TABLE V: Number of test images per subject for each FCFD experiment. All test images were tested both against a probe from the same day and a probe from a different day.

Experiment	Number of Images per Subject	
	Easy	Complete
Lighting	4	14
Expression	8	13
Angles	8	12
Scales	3	10
All	58	252

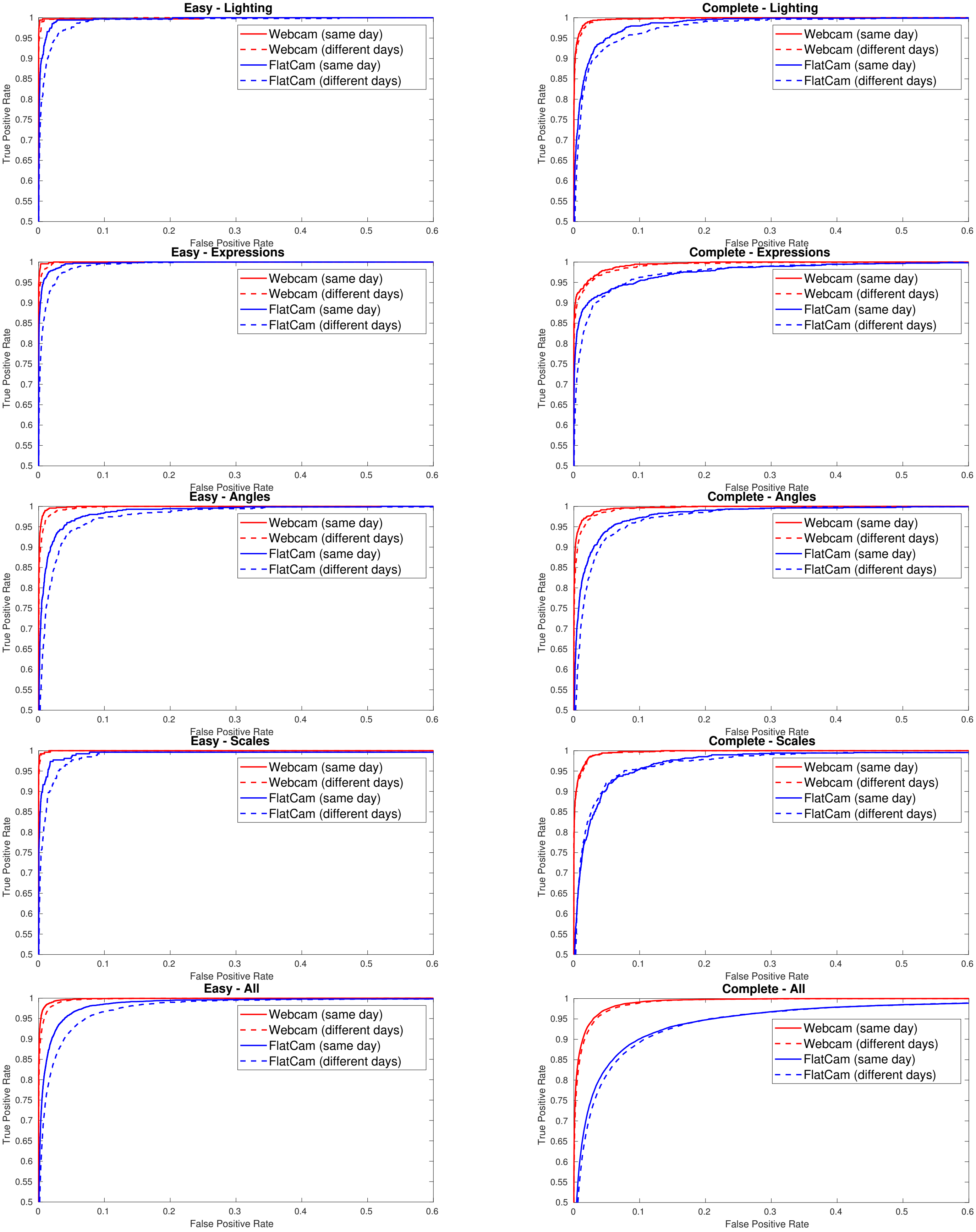


Fig. 10: ROC curves for verification results on the FCFD dataset for different scenarios

TABLE VI: True positive rates for the different FCFD test scenarios at fixed false positive rates.

Set	**Variation**	**Same Day**				**Different Days**			
		Webcam		**FlatCam**		**Webcam**		**FlatCam**	
		FPR=1%	FPR=0.1%	FPR=1%	FPR=0.1%	FPR=1%	FPR=0.1%	FPR=1%	FPR=0.1%
Easy	Lighting	99.72%	98.01%	94.03%	77.27%	99.43%	94.89%	86.74%	63.26%
Easy	Expression	99.57%	96.59%	96.02%	85.37%	98.41%	92.50%	87.95%	66.36%
Easy	Angles	99.01%	89.06%	83.66%	57.67%	96.16%	83.81%	72.02%	40.77%
Easy	Scales	99.62%	98.11%	92.42%	78.41%	99.62%	95.45%	83.33%	59.47%
Easy	All	98.02%	89.81%	82.64%	55.25%	95.74%	83.94%	73.94%	43.48%
Complete	Lighting	96.72%	86.11%	79.80%	57.83%	95.34%	84.43%	74.77%	42.39%
Complete	Expression	93.18%	87.33%	86.45%	73.25%	91.82%	84.17%	78.26%	57.20%
Complete	Angles	95.64%	83.81%	79.45%	51.61%	93.28%	78.12%	69.22%	38.35%
Complete	Scales	94.43%	83.52%	70.68%	47.73%	93.75%	84.20%	72.39%	37.61%
Complete	All	87.74%	69.53%	61.38%	34.09%	85.79%	65.18%	57.16%	27.53%

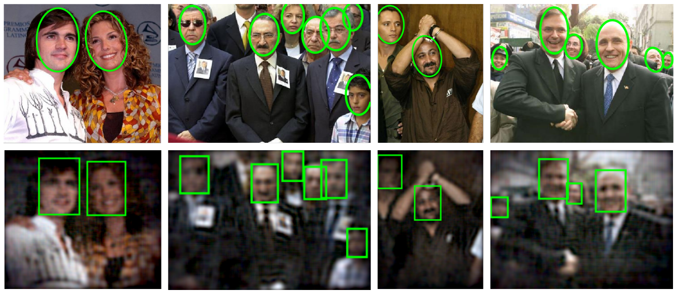


Fig. 11: Original elliptical annotations for the FDDB dataset (top) and sample bounding boxes detected by the Faster R-CNN detector on its lensless counterpart (bottom).

difference between the accuracies in the Easy and Complete sets for the FlatCam also highlight how subjects looking down, dramatic facial expressions, and the lack of a frontal light source (variations only included in the complete set) greatly hurt the performance of lensless face verification. This may be because the lack of a frontal light source, as well as looking down (when most light sources are above the head), cause shadowed regions on the face, which the FlatCam may not effectively sense due to its lower dynamic range as described in section III. Dramatic expressions, on the other hand, also have a big effect on standard face verification, which means it is tough for face verification in general. The large decrease in performance between the Easy and Complete tests for the Scale variation may be because bringing the subject closer to the camera sharpened the angles of the side lamps used in the Complete set.

B. Experiments on Standard Face Datasets

We also run experiments on lensless versions of the Face Detection Data Set and Benchmark (FDDB) dataset [68] and the Labeled Faces in the Wild (LFW) dataset [53], [54], two benchmark test datasets, which we obtain by displaying the images of these datasets on a screen and capturing them with our lensless system. While these are not images of real physical faces, results from these experiments can be compared with those reported in other works for face detection and verification on these datasets. For these experiments, the simulated training images were simulated with noise N such that $\frac{||\Phi_L X \Phi_R||}{||N||} = 20$ in Eq. 1 and with $\lambda = 3 \times 10^{-4}$ in Eq. 3.

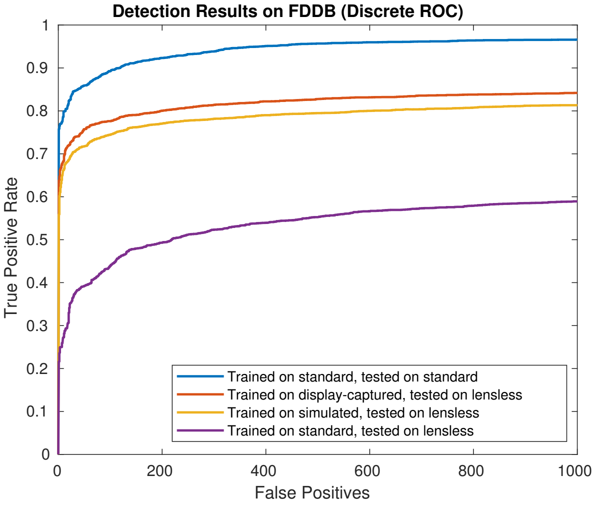


Fig. 12: Discrete ROC for detection on FDDB. Using a lensless imaging system generally decreases the accuracy by around 13% when trained on screen captured images. Training on simulated images further decreases the accuracy by around 3%. Training on standard images when testing on lensless images yields much lower accuracy rates.

1) Face Detection on FDDB: For detection, we test on FDDB using the provided evaluation code. This dataset contains elliptical annotations for 5171 faces in 2845 images and is commonly used as a test dataset for face detection algorithms. In addition to testing the models trained on the different types of training data on the lensless version of FDDB, we also test the model trained on standard images on the standard FDDB for comparison.

We report the discrete results [68] for FDDB, as shown in Fig. 12. The results are consistent with those from the FCFD tests. Again, it is necessary to train Faster R-CNN on lensless images, whether display-captured or simulated, when testing on lensless images. Doing so increases the accuracy by approximately 26% compared to using a model trained simply on standard images. Second, we show that generally, detection on the lensless images yields about a 13% decrease in accuracy compared to performing detection on standard images. Lastly, using screen captured images provides an increased accuracy of around 3% compared to simulated images. The results

TABLE VII: Accuracy rates for different training and test schemes on the LFW dataset.

Training Images	Test Images	Accuracy Rate
Standard	Standard	0.9758 ± 0.0033
Standard	Lensless	0.8882 ± 0.0034
Display-captured	Lensless	0.9380 ± 0.0036
Simulated	Lensless	0.9183 ± 0.0046

suggest that one can still attain reasonable accuracy for face detection on lensless images as long as one trains the system on lensless images. Sample bounding boxes identified by the display-captured detector are shown in Fig. 11.

2) Face Verification on LFW: For face verification, we run experiments on the LFW dataset, which contains 13,233 images of 5749 subjects [53], [54]. As with detection, we test the three CNN models trained with the different types of training data and also test on the standard LFW dataset. For the pre-processing step of cropping the face patches, we use the bounding boxes obtained using the standard Faster R-CNN detector on standard LFW images for both standard and lensless LFW images for a fair comparison.

Since the CNNs we use for face verification are trained on data outside of the LFW dataset, we operate in LFW's "unrestricted with labeled outside data" protocol [54]. The standard way of reporting results in this protocol is to present accuracy rates on their provided 10-fold cross validation sets. We use the 9 folds to determine the best threshold to apply on the feature distances and apply this threshold on the 10th fold. Our results are reported in Table VII.

Similar to detection, there is a small performance decrease (around 3.8%) when using lensless images instead of standard images. The importance of training with lensless images is once again highlighted in this experiment, and using simulated images again slightly decreases the accuracy compared to using display-captured images.

C. Discussion

From all three test datasets, there is the common trend that, when testing on lensless images, training on standard images yields poor accuracy and training on display-captured lensless images yields much improved accuracy. Using simulated images gives lower accuracy than display-captured images, which we note may be due to our current simulation procedure. A more accurate simulation may improve these results. However, currently, simulating training images still achieves better performance than using standard images. If one cannot obtain display-captured images due to physical or time constraints, simulating training data is a tractable and efficient choice to obtain reasonable performance.

The results on our captured FCFD are the most accurate indicators for real applications, since it is the only dataset with lensless images of real human faces. From our results, we observe that we obtain the best verification accuracy when the subjects are generally facing towards the camera in a well-lit environment. The ideal lighting condition is when the illumination is sufficiently diffuse or if there is a source of lighting from the front of the subject's face (to minimize the effects of shadows). Given these two scenarios, the subject can make changes in facial expressions without decreasing the performance greatly. On the other hand, faces angled downwards are very difficult for lensless verification. These statements are also true for lens-based cameras, but the effect is much more dramatic for lensless cameras possibly since they are more sensitive to angles and lighting due to their lower dynamic range.

In the Supplementary Material, we show that the networks we train using one FlatCam device can be used on images captured by other FlatCam devices (with different mask patterns and mask misalignments) without any decrease in performance. This shows the generalizability of face detection and verification with FlatCam across devices.

Other recent methods for performing face detection and verification include specific and sophisticated techniques such as using facial landmarks or using a larger number of CNNs. However, we choose more general deep learning methods to show that one does not need heavy machinery to achieve reasonable performance. Of course, additional tactics can be incorporated into the general methods to better fit a specific application or further improve the performance reported here.

While performing face detection and verification on lensless images yields lower performance than on standard images, lensless imaging systems, such as FlatCam, can have a much lower cost and thinner form factor. This may be an attractive tradeoff in application scenarios with stringent geometric, size, and cost constraints, in which standard lens-based cameras need to be replaced with lensless systems.

Moreover, some applications may have simpler settings and less variations than those present in our FCFD dataset, which might lead to much higher accuracy of lensless systems in those scenarios. In addition, this is also the first-ever study on lensless face verification, and as has been seen in the field of computer vision, additional research may soon provide much greater accuracy for face detection and verification with lensless cameras.

VIII. Conclusions

In this work, we have evaluated the performance of face detection and face verification algorithms on images captured by FlatCam to study whether these tasks can be performed with a lensless imaging system rather than a conventional camera. We have observed that, despite the minor drop in performance when using lensless images, the accuracy may still be sufficient for many applications. Many of the methods we used were rather simple, and researchers are continuously developing more sophisticated methods to improve results for these tasks. Thus, it is possible to transfer those complex methods to lensless imaging to improve our results. Ultimately, shrinking the performance gap between lensless and lens-based inference is now an open problem. We also note that, while the lensless imaging community is working to make imaging systems thinner and cheaper, there is also ongoing work by other researchers in making deep learning more efficient and scalable [70], [71]. Such work will culminate in small, efficient, and inexpensive devices that offer an attractive performance tradeoff.

Face detection and verification are only two examples in a rich list of interesting inference tasks in computer vision. Now that we have proved that lensless cameras can succeed on these tasks, it appears likely that they will also succeed on other tasks such as object detection, gesture recognition, and beyond.

We invite researchers to test their face verification algorithms on our FlatCam Face Dataset (FCFD), which will aid in evaluating the generalizability of existing algorithms to another image domain and facilitate applications with thin lensless imaging systems. However, we do note that the FCFD is only based on the FlatCam, one type of lensless imager, which has characteristics that may be different from other types of lensless imaging systems. We also recommend that users who are interested in performing face detection and verification under stringent cost or size constraints consider performing these tasks with thin lensless imaging systems as we have.

References

[1] TechInsights. (2018, Mar) Cost comparison – Samsung Galaxy S9+, Samsung Galaxy Note 8, Samsung Galaxy 8+, Apple iPhone 8+, Apple iPhone X. Accessed 2018-11-01. [Online]. Available: http://www.techinsights.com/about-techinsights/overview/blog/cost-comparison-samsung-galaxy-s9-plus-samsung-galaxy-note-8-samsung-galaxy-8

[2] M. S. Asif, A. Ayremlou, A. Sankaranarayanan, A. Veeraraghavan, and R. G. Baraniuk, "Flatcam: Thin, lensless cameras using coded aperture and computation," *IEEE Trans. Comput. Imag.*, vol. 3, no. 3, pp. 384–397, Sept 2017.

[3] J. K. Adams, V. Boominathan, B. W. Avants, D. G. Vercosa, F. Ye, R. G. Baraniuk, J. T. Robinson, and A. Veeraraghavan, "Single-frame 3d fluorescence microscopy with ultraminiature lensless flatscope," *Sci. Advances*, vol. 3, no. 12, p. e1701548, 2017.

[4] D. G. Stork and P. R. Gill, "Lensless ultra-miniature cmos computational imagers and sensors," in *Int. Conf. Sensor Technol. Appl.* Citeseer, 2013, pp. 186–190.

[5] A. Ozcan and E. McLeod, "Lensless imaging and sensing," *Annu. Rev. Biomed. Eng.*, vol. 18, pp. 77–102, 2016.

[6] V. Boominathan, J. K. Adams, M. S. Asif, B. W. Avants, J. T. Robinson, R. G. Baraniuk, A. C. Sankaranarayanan, and A. Veeraraghavan, "Lensless imaging: A computational renaissance," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 23–35, 2016.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2001.

[8] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," 2010.

[9] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: past, present and future," *Comput. Vision Image Understanding*, vol. 138, pp. 1–24, 2015.

[10] X. Sun, P. Wu, and S. C. Hoi, "Face detection using deep learning: An improved faster rcnn approach," *arXiv preprint arXiv:1701.08289*, 2017.

[11] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," in *2017 12th IEEE Int. Conf. Automat. Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 650–657.

[12] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *arXiv preprint arXiv:1603.01249*, 2016.

[13] M. Najibi, P. Samangouei, R. Chellappa, and L. Davis, "Ssh: Single stage headless face detector," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2017, pp. 4875–4884.

[14] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Faceness-net: Face detection through deep facial part responses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, 2018.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[16] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[17] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3d and multi-modal 3d+2d face recognition," *Comput. Vision Image Understanding*, vol. 101, no. 1, pp. 1–15, 2006.

[18] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: A survey," *Pattern Recognition*, vol. 39, no. 9, pp. 1725–1745, 2006.

[19] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Brit. Mach. Vision Conf.*, September 2015, pp. 41.1–41.12.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2015, pp. 815–823.

[21] Y. Sun, D. Liang, X. Wang, and X. Tang, "Deepid3: Face recognition with very deep neural networks," *arXiv preprint arXiv:1502.00873*, 2015.

[22] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2014, pp. 1701–1708.

[23] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 964–975, Feb 2018.

[24] C. Xiong, L. Liu, X. Zhao, S. Yan, and T. K. Kim, "Convolutional fusion network for face verification in the wild," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 3, pp. 517–528, March 2016.

[25] S. Ouyang, T. Hospedales, Y.-Z. Song, X. Li, C. C. Loy, and X. Wang, "A survey on heterogeneous face recognition: Sketch, infra-red, 3d and low-resolution," *Image Vision Comput.*, vol. 56, pp. 28–48, 2016.

[26] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, "Recent advances in visual and infrared face recognitiona review," *Comput. Vision Image Understanding*, vol. 97, no. 1, pp. 103–135, 2005.

[27] P. Buddharaju, I. T. Pavlidis, P. Tsiamyrtzis, and M. Bazakos, "Physiology-based face recognition in the thermal infrared spectrum," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 613–626, 2007.

[28] W. Burgin, C. Pantofaru, and W. D. Smart, "Using depth information to improve face detection," in *Proc. 6th Int. Conf. Human-Robot Interaction*. ACM, 2011, pp. 119–120.

[29] S. Eisebitt, J. Lüning, W. Schlotter, M. Lörgen, O. Hellwig, W. Eberhardt, and J. Stöhr, "Lensless imaging of magnetic nanostructures by x-ray spectro-holography," *Nature*, vol. 432, no. 7019, pp. 885–888, 2004.

[30] E. Caroli, J. Stephen, G. Di Cocco, L. Natalucci, and A. Spizzichino, "Coded aperture imaging in x- and gamma-ray astronomy," *Space Sci. Rev.*, vol. 45, no. 3-4, pp. 349–403, 1987.

[31] G. Huang, H. Jiang, K. Matthews, and P. Wilford, "Lensless imaging by compressive sensing," in *2013 20th IEEE Int. Conf. Image Process. (ICIP)*. IEEE, 2013, pp. 2101–2105.

[32] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 83–91, 2008.

[33] A. Zomet and S. K. Nayar, "Lensless imaging with a controllable aperture," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 1. IEEE, 2006, pp. 339–346.

[34] W. Chi and N. George, "Optical imaging with phase-coded aperture," *Optics express*, vol. 19, no. 5, pp. 4294–4300, 2011.

[35] M. J. DeWeert and B. P. Farm, "Lensless coded-aperture imaging with separable doubly-toeplitz masks," *Opt. Eng.*, vol. 54, no. 2, p. 023102, 2015.

[36] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[37] G. Kim, S. Kapetanovic, R. Palmer, and R. Menon, "Lensless-camera based machine learning for image classification," *arXiv preprint arXiv:1709.00408*, 2017.

[38] M. Monjur, L. Spinoulas, P. R. Gill, and D. G. Stork, "Ultra-miniature, computationally efficient diffractive visual-bar-position sensor," in *Proc. 9th Int. Conf. Sensor Technol. Appl. (SENSORCOMM 2015)*, 2015, pp. 24–29.

[39] A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," *Optica*, vol. 4, no. 9, pp. 1117–1125, 2017.

[40] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Advances Neural Inf. Process. Syst.*, 2017, pp. 1772–1783.

[41] S. Li, M. Deng, J. Lee, A. Sinha, and G. Barbastathis, "Imaging through glass diffusers using densely connected convolutional networks," *Optica*, vol. 5, no. 7, pp. 803–813, 2018.

[42] Y. Li, Y. Xue, and L. Tian, "Deep speckle correlation: a deep learning approach toward scalable imaging through scattering media," *Optica*, vol. 5, no. 10, pp. 1181–1190, 2018.

[43] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2016, pp. 449–458.

[44] J.-H. R. Chang, C.-L. Li, B. Poczos, B. V. Kumar, and A. C. Sankaranarayanan, "One network to solve them all-solving linear inverse problems using deep projection models." in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 5889–5898.

[45] A. Mousavi, G. Dasarathy, and R. G. Baraniuk, "Deepcodec: Adaptive sensing and recovery via deep convolutional neural networks," in *55th Annu. Allerton Conf. Commun., Control, Comput.*, Oct 2017, pp. 744–744.

[46] S. R. Gottesman and E. Fenimore, "New family of binary arrays for coded aperture imaging," *Appl. Opt.*, vol. 28, no. 20, pp. 4344–4352, 1989.

[47] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1090–1104, 2000.

[48] A. M. Martinez and R. Benavente, "The ar face database," *CVC Tech. Rep.*, 1998.

[49] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, vol. 1. IEEE, 2005, pp. 947–954.

[50] A. Georghiades, P. Belhumeur, and D. Kriegman, "Yale face database," *Center Comput. Vision and Contr. at Yale University, http://cvc. yale. edu/projects/yalefaces/yalefa*, vol. 2, 1997.

[51] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, 2001.

[52] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "The cmu multi-pose, illumination, and expression (multi-pie) face database," *CMU Robotics Institute. TR-07-08, Tech. Rep.*, 2007.

[53] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[54] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[55] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[56] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The megaface benchmark: 1 million faces for recognition at scale," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2016, pp. 4873–4882.

[57] J. Tan, V. Boominathan, A. Veeraraghavan, and R. Baraniuk, "Flat focus: depth of field analysis for the flatcam lensless imaging system," in *2017 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. IEEE, 2017, pp. 6473–6477.

[58] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2016.

[59] P. M. R. Martin Koestinger, Paul Wohlhart and H. Bischof, "Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization," in *Proc. 1st IEEE Int. Workshop Benchmarking Facial Image Anal. Technol.*, 2011.

[60] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[61] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conf. Comput. Vision*. Springer, 2016, pp. 21–37.

[62] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec 2015, pp. 1440–1448.

[63] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[64] J.-C. Chen, V. M. Patel, and R. Chellappa, "Unconstrained face verification using deep cnn features," in *2016 IEEE Winter Conf. Appl. Comput. Vision (WACV)*. IEEE, 2016, pp. 1–9.

[65] N. Crosswhite, J. Byrne, C. Stauffer, O. Parkhi, Q. Cao, and A. Zisserman, "Template adaptation for face verification and identification," in *2017 12th IEEE Int. Conf. Automat. Face Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 1–8.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[67] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proc. 23rd ACM Int. Conf. Multimedia*. ACM, 2015, pp. 689–692.

[68] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[69] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[70] Y. Lin, C. Sakr, Y. Kim, and N. Shanbhag, "Predictivenet: An energy-efficient convolutional neural network via zero prediction," in *2017 IEEE Int. Symp. Circuits Syst. (ISCAS)*. IEEE, 2017, pp. 1–4.

[71] R. Spring and A. Shrivastava, "Scalable and sustainable deep learning via randomized hashing," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. ACM, 2017, pp. 445–454.