Efficient Neural Vision Systems Based on Convolutional Image Acquisition

Pedram Pad, Simon Narduzzi, Clément Kündig,
Engin Türetken, Siavash A. Bigdeli, L. Andrea Dunbar

Centre Suisse d'Électronique et de Microtechnique (CSEM), Neuchâtel, Switzerland
www.csem.ch

Abstract

Despite the substantial progress made in deep learning in recent years, advanced approaches remain computationally intensive. The trade-off between accuracy and computation time and energy limits their use in real-time applications on low power and other resource-constrained systems. In this paper, we tackle this fundamental challenge by introducing a hybrid optical-digital implementation of a convolutional neural network (CNN) based on engineering of the point spread function (PSF) of an optical imaging system. This is done by coding an imaging aperture such that its PSF replicates a large convolution kernel of the first layer of a pre-trained CNN. As the convolution takes place in the optical domain, it has zero cost in terms of energy consumption and has zero latency independent of the kernel size. Experimental results on two datasets demonstrate that our approach yields more than two orders of magnitude reduction in the computational cost while achieving near-state-of-the-art accuracy, or equivalently, better accuracy at the same computational cost.

1. Introduction

In recent years, convolutional neural networks (CNNs) have proven to be very powerful for various vision applications such as image classification and object detection, among many others [30, 13]. However, practical implementations still remain in the order of giga multiplication-addition operations (MAdds) [32, 36] despite the significant effort that has been put into lowering their computational cost [15]. This poses a major barrier in many embedded intelligence applications with ultra-low power, small form factor or low-cost requirements which impose strong constraints on the available computational resources to run them in real-time. Such constraints are present in an increasing number of application domains such as internet of things (IoT), smart sensors and quality control systems, which creates a strong incentive to develop new approaches that can deliver high accuracy at a low power budget and with limited computational resources.

Optical systems provide efficient computing capabilities thanks to their inherent parallelism and extremely high speed while effectively consuming no power [37]. Imaging an object through an optical system can be modelled as its convolution with its PSF which is shift invariant under certain assumptions. Engineering this function has recently become a widespread practice in numerous applications such as monocular depth estimation [14], de-blurring [21], template matching [18], and privacy preservation through random coded apertures in human video sequences [38]. This approach has also been proposed as a way to efficiently run neural networks in the optical domain [4]. However, this work is purely a conceptual design of an optical system to perform matrix multiplication and no physical implementation of it in practice has been proposed. Recently, new practical ideas have been proposed to partially or fully outsource the neural network computations from the processing unit to the optical frontend [22, 6, 10]. These techniques rely on phase masks, and hence, their systems require a coherent monochromatic light source to function. This is a serious drawback as natural scenes emit incoherent and polychromatic light. In such systems this produces a considerable amount of chromatic aberration that dramatically degrades their performance. As a result, these approaches are not readily applicable to general-purpose computer vision tasks in practical applications. For the sake of completeness, it should be mentioned that there have been efforts to perform some basic image processing on the sensor pixel before photon-to-electron conversion which are out of the scope of this work [8].

In this work, we propose a generic approach for optical convolutions based on amplitude-varying masks to address the challenge of processing incoherent and broadband light that exists in natural scenes. More specifically, we design a compact optical system, made up of an amplitude-only transmittance mask and double lenses. The physical mask

is obtained by transcribing the pre-trained weights of a digital convolutional layer onto it such that the PSF function of the optical system as a whole closely approximates the convolution kernel. The resulting image acquired by an image sensor is then given as an input to the remaining layers of the network, which are designed to be low-complexity to keep the computational resources required to run it at a minimum. Our hybrid optical-digital approach is therefore particularly suitable for real-time embedded inference applications, where low-power consumption and low-latency are of great importance.

In order to get the most out of the optical frontend, we propose a shallow neural network architecture comprised of a large first convolution layer (in terms of kernel size) for a high modelling capacity, followed by a small number of layers to be executed in the digital domain. In contrast to most CNN architectures that use many small kernels (e.g., 3×3 and 5×5), our optical convolution layer design allows scaling kernels to be even larger than the input in size. In fact, optimizing the integration of the kernel onto the transmittance mask leads us to choose a single kernel that is several times larger than the input. This single kernel is expected to have the capacity to learn a high number of distinct and informative features that would otherwise be achieved only by using many smaller ones. Our proposed setup remains easy-to-fabricate using current printing technologies and does not bare significant cost to the production of cameras.

Filtering out irrelevant information early on in the optical domain leads to extremely light-weight digital architectures and is akin to the mammalian vision systems, whose retinal ganglion cells extract the features natural scenes with their receptive field [24]. Retinal cells also capture scene information in a transformation domain (Gabor-like wavelets [20]) instead of recording a grid of pixel intensities, which have large redundancy and less information [28, 26].

In the following Section, we first explain the general concept of performing a spatial convolution in the optical domain and then present key details about our specific implementation. Section 3 describes the details of our prof-of-concept implementation. We discuss our techniques for end-to-end training of the parameters of our network in Section 4. In Section 5, we show the performance of our proposed approach for optical character and hand-gesture recognition tasks and show competitiveness compared to other state-of-the-art methods. Our hybrid approach requires only a fraction of the memory and computational resources required by state-of-the-art algorithms while achieving similar accuracy. We provide additional technical details for our optical setup in Appendix I.

2. Spatial convolution in optical domain

In this section, we propose an optical system which performs the convolution of the scene with an arbitrary predefined kernel. The optical basis for geometrical transformation of light through a system of lenses and how the presented setup works is presented in Appendix I for the sake of readability of the paper.

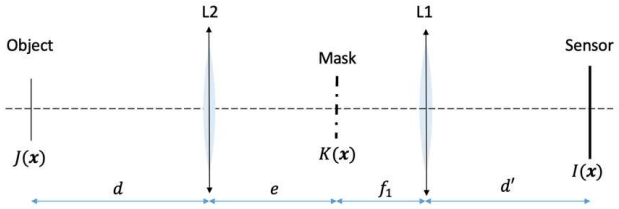


Figure 1: Double lens optical setup for spatial convolution.

Assume that our scene is defined by function $J(\mathbf{x})$ from $\mathbb{R}^2$ to $\mathbb{R}^+$ ($\mathbb{R}^+$ is the set of non-negative real numbers which are the luminosity of light at position $\mathbf{x}$)[1]. Also, assume that $K(\mathbf{x})$ from $\mathbb{R}^2$ to $[0, 1]$ is a spatially coded transmission mask which means that of the light that arrives at position $\mathbf{x}$ on the mask, $K(\mathbf{x})$ is the portion of it that passes through the aperture mask and the rest is absorbed or reflected. Now, having the optical setup depicted in Figure 1, we have the following relationship between what we receive on the sensor plane and the scene:

$$I(\mathbf{x}) = \left(J(\alpha\,\cdot) * \gamma K(\gamma\,\cdot) T(\gamma\,\cdot)\right)(-\mathbf{x}) \tag{1}$$

where $*$ indicates the standard 2-dimensional convolution, $T(\mathbf{u}) = \frac{1}{2\pi}(1 + \|\mathbf{u}\|_2^2)^{-\frac{3}{2}}$ in which $\|\cdot\|_2$ is the Euclidean L_2-norm and

$$\alpha = \frac{de}{f_1}\left(\frac{1}{d} + \frac{1}{e} - \frac{1}{f_2}\right), \tag{2}$$
$$\gamma = \left(\frac{de}{f_1}\left(\frac{1}{d} + \frac{1}{e} - \frac{1}{f_2}\right)\left(1 - \frac{d'}{f_1}\right) + \left(1 - \frac{d}{f_2}\right)\right)^{-1}$$

where f_1 and f_2 are focal lengths of the lenses L1 and L2, respectively. Due to physical constraints, $\|\mathbf{u}\|_2$ is usually small so that $T(\mathbf{u})$ is nearly a constant function.

This setup can be used to make the convolution in the optical domain. This convolution consumes effectively zero time and energy irrespective of size. Based on this scheme, we perform the first layer of our CNN in the optical domain by realizing the trained filters of the first layer as a spatially coded mask. Thus the first layer of the CNN, which is usually the most computationally expensive layer, is no longer carried out in the digital domain and can take advantage of the optical domain. Moreover, this idea can be used in other image processing applications like compression and denoising [7, 27, 25, 3] since they start with filtering the images to make them faster and more energy efficient. Notice that the wavelet transform can also be performed using this setup by printing different wavelet filters on different locations of

[1] Please refer to Equation 7 in Appendix I for more explanation

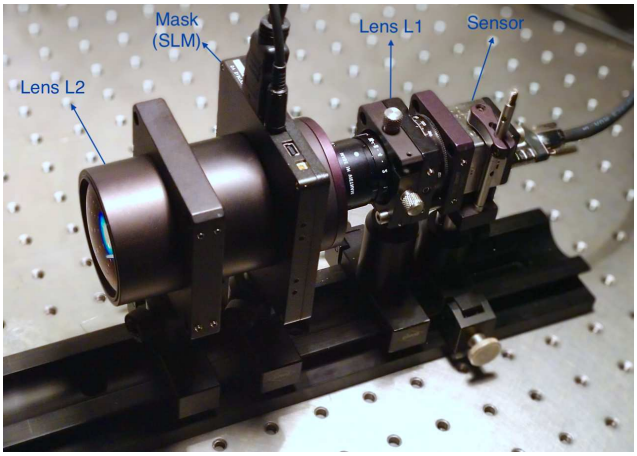
the mask plane. In general, $K(\mathbf{x})$ can be the concatenation of several finite impulse response (FIR) filters with proper spacing between them. Also, it can yield more accurate results in processing systems with high round-off error.

Figure 2(a) shows an experimental setup implementing the proposed imaging system described above (corresponding to the schematic drawing in Figure 1). The mask, lens and the image sensor are mounted in optical baffle tubes, placed in a cage system to ensure alignment and reduced light pollution. The output signal is captured by an off-the-shelf camera interfaced with a computer. Here, we used a spatial light modulator (SLM), which is a programable transmission mask, to create the mask $K(\mathbf{x})$ containing 9 kernels of size 3×3 trained for the MNIST application[19]. The spacing between the kernels are designed such that the outcome of the 9 convolutions do not overlap with each other. The target scene, the kernel mask and the recorded signal on the image sensor are shown in Figures 2(b), 2(c) and 2(d), respectively.

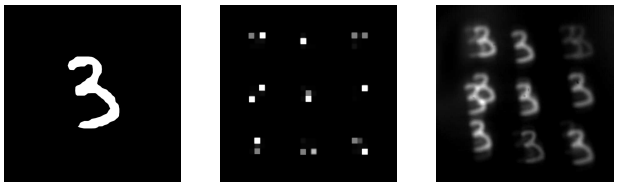
We can include a layer at the beginning of our network with many parameters using the optical convolution, which comes with effectively zero cost (in terms of time and energy) irrespective of its kernel size (number of parameters). The outcome of this convolution is then the first feature map input to the rest of the neural network. This scheme leads to a more general, and potentially richer, feature map encodings given that the conventional convolutional layer (i.e. having several small kernels with zero spacing between them) is a special case of this big kernel. A sample of such a kernel and its corresponding output for the input image of Figure 2(e) is demonstrated in Figures 2(f) and 2(g), respectively. Due to these attractive properties, we use the optical convolutional layer in our neural networks and demonstrate near-state-of-the-art performance in our experiments.

3. Design and implementation

In this section, we present the pipeline of designing and implementing a system based on the optical convolution module along with the specifications of our setup. First, we design the neural network architecture in which the first layer is a single huge kernel (instead of a series of for example 3×3 or 5×5 kernels which are commonly used). The kernel size depends on the physical size of the optical system and the mask fabrication resolution. The activation function after the first big convolution depends on the image sensor measurement strategy. Often the image sensors read the pixel values in a linear scale which is equivalent to a linear activation function in our case. However, there are image sensors (such as ERGO [29]) in which the sensor reads the logarithm of the pixel values directly. Thus, using this type of image sensors, we gain an additional nonlinearity and potentially more capacity without any computational cost.



(a)



(b) (c) (d)

(e) (f) (g)

Figure 2: (a) Photograph of the complete experimental setup, (b) Sample image from the MNIST dataset displayed on a LCD screen in front of the camera. (c) Physical mask created from 9 trained 3×3 kernels tiled side by side. (d) Convolution between the input image and the patterns on the mask captured at the image sensor. (e) Sample image from the MNIST dataset displayed on a LCD screen in front of the camera. (f) Physical mask created from 1 trained 240×240 kernel. (g) Convolution between the input image and the patterns on the mask captured at the image sensor.

In our case, we used a 240×240 kernel (the resolution of dots on the mask is $36\,\mu\text{m}$, which results in a total side length of $8.64\,\text{mm}$). For the rest of the network, we used a 4-layer fully connected perceptron with 256 neurons in each layer (the number of output neurons depends on the application). We chose this network since it is implemented as an application specific integrated system (ASIC) by Syntiant company in a very efficient manner [1] giving a power consumption of $150\,\mu\text{W}$. The input to this network has 1600

dimensions. Thus, we down-sample the result of the optical convolution into a 40×40 image in order to feed it to the network. For the image sensor, both linear and logarithmic ones are implemented. After designing the network in digital domain, we train it for a given task. It is important to consider that as this system works with incoherent light (which has only a well-defined amplitude and not phase), the kernel entries can take values between 0 and 1; 0 means complete blocking of light and 1 means complete transmission of light. In our case we trained them for three datasets of optical character recognition (OCR) application which are explained in the next section.

Once the mask and the network are trained, we produce the mask as an optical element and load the rest of the network to our processing unit (see Figure 3a). There are several ways to produce the mask such as aerosol-jet printing on glass [39], high-resolution inkjet printing [33] and using a spatial light modulator (SLM) [12]. We used the SLM since its electronically programable mask and suitable for experimental tasks. However, in a final system an industrial low cost fixed mask can be used. The concept of SLM functionality is the same as a liquid crystal display (LCD) in which the amount of light that can be passed through each pixel is controlled by changing the polarization of the incoming light by passing it through an electromagnetic field. In our case, the mask pixel size is $36\,\mu\mathrm{m}$, as mentioned above. However, with aerosol-jet printing it can be reduced down to $10\,\mu\mathrm{m}$ which means substantially more parameters to train (more capacity for the network) or much smaller optical setup for the same capacity. It should be noted that based on Equation 1, we derive the right values for f_1, f_2, e, d and d' for which the input image size, input image pixel size, kernel size, kernel pixel size, total sensor size and sensor pixel pitch match together.

It should be further noted that optical distortions can be implemented as data augmentation methods during the training phase in order to increase the robustness. Nevertheless, after implementing the system, one can fine-tune it by collecting more practical data. Notice that having the recorded data on the sensor is enough for fine-tuning the part of the network which is performed by the processing unit. However, fine-tuning the mask is also possible by having a programable mask (e.g. SLM) and knowing the exact scene in front of the camera. This can be done through displaying known data in front of the camera (see Figure 3b). This method can be used from the beginning to train the mask and the network (evolution from random mask and weights to the ones that perform a meaningful task).

4. Neural network architecture and training

In this section, we describe our proposed system, and the training strategies for our networks.

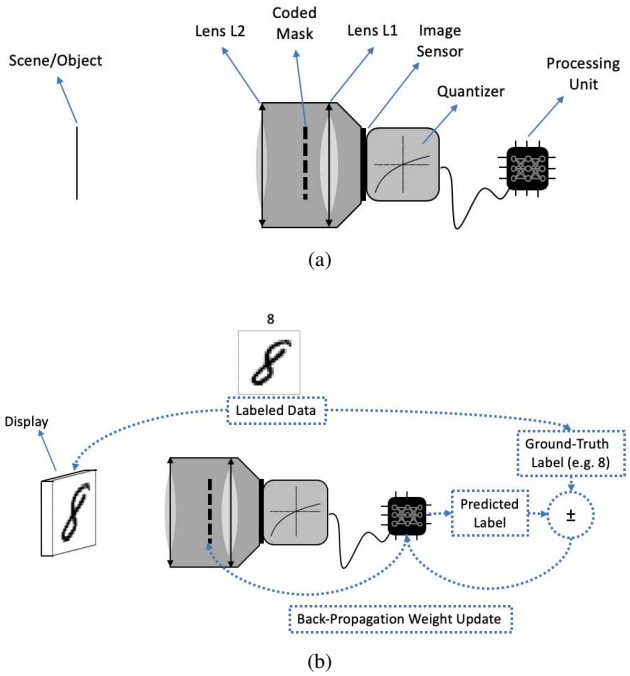


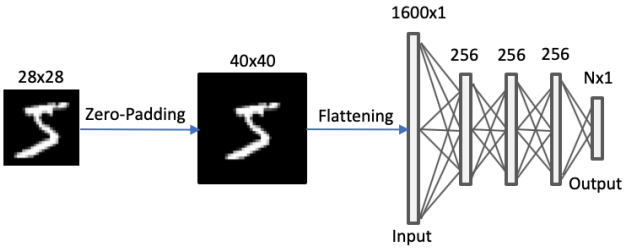
Figure 3: (a) The proposed vision system with a computation-free convolution in optical domain, a computation-free activation function in the image sensor and the processing unit containing a standard neural network. (b) The setup for training or fine-tuning the weights of the network (and the coded mask) after implementing the system in hardware.

4.1. System architecture

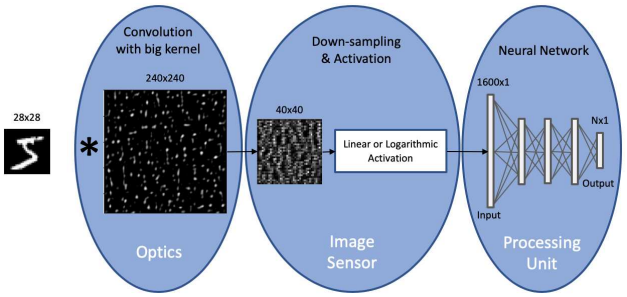
Our goal was to design an ultra-efficient classification system for the OCR application. Therefore, we selected the ultra-low power Syntiant NDP101 Neural Decision Processor™[1] as the processing unit in combination with the ultra-low power image sensor ERGO [29]. The NDP101 contains a neural processing engine consisting of a perceptron neural network with 1600 input vector, 3 fully connected layers each with 256 neurons with rectified linear unit (ReLu) activation function and one fully connected layer with linear activation for the output units. Beside being ultra-low power, the ERGO image sensor enables us to switch between the linear or logarithmic quantification of the pixel values.

Based on the physical characteristics of our setup, the size of our optical convolution kernel is 240×240. In order to have 1600-dimensional input vector to feed into the perceptron network, we read a grid of 40×40 pixels on the sensor. This is implemented by sub-sampling the image in our hardware, and using strided-convolutions in our training.

In order to assess the effect of our system's components



(a)



(b)

Figure 4: (a) Baseline architecture (Syntiant perceptron): A fully-connected network with 3 hidden layers of 256 neurons. (b) Architecture in which a big optical convolution with linear activation (OptConv+Perc) or logarithmic activation (OptConv+Log+Perc) is placed before the perceptron network.

independently, we start with a baseline model which is only the aforementioned perceptron neural network (Figure 4a). After that, we study the performance of the system after adding the optical convolution. For the quantization strategy of the image sensor, we first consider the standard linear quantization and then the logarithmic quantization (Figure 4b).

4.2. Training

The models were trained on some subsets of the EMNIST dataset [9], namely EMNIST-Digits, EMNIST-Letters and EMNIST-Balanced. We performed data augmentation using random rotations between -10° and 10° for input images. Additionally, we added Gaussian noise with a standard deviation of 1.5 with the probability of 30%. With the same probability, we also blurred the images using a Gaussian kernel, where we selected its bandwidth randomly between 1 and 4. Each model was trained with the Adam optimizer with initial learning rate of 10^{-5} for 1000 epochs. During the training, 5% of the training set was separated as the validation set and the network with the best validation accuracy was selected for testing.

4.2.1 Training with logarithmic activation

Since the network (including the optical convolution) is relatively shallow, training the network with logarithmic activation after the convolution often ends in a non-performant local minima. To avoid this problem, we first train a model with linear activation. After convergence, we switch the linear activation to the logarithmic one. This sudden change of activation can cause an imbalance in training, therefore we use scaling transformation to reduce this effect.

According to the Taylor expansion theorem, for x being in a vicinity of 0, we have

$$\log(1+x) \approx x. \tag{3}$$

Thus, buy scaling the input values with a large number a, we have

$$a \log\left(1+\frac{x}{a}\right) \approx x. \tag{4}$$

Therefore, after switching the activation from linear to logarithmic, we divide the weights of the convolution layer by a large factor and multiply the weights of the first layer of the perceptron network by the same factor. In practice, we found $a = 1000$ to be large enough to obtain a stable transition. The additive constant 1 inside the logarithm is set as the bias of the convolution layer. After this re-configuration, we can train the network using conventional techniques.

Light throughput regularization term. Having a mask with weight values closer to 1 results in more throughput of light and therefore higher signal to noise ratio on the image sensor. In order to come up with such kernels, we can add the term $\frac{\epsilon}{\|K(\mathbf{x})\|_1}$, where ϵ is a positive scalar and $\|\cdot\|_1$ is the standard L_1-norm, to the cost function used for training of the network. Thus, during training the kernels are more favorable that have higher L_1-norms (sum of entries). Experiments showed that this term can result in kernels with 10 times higher light throughput without any significant change of accuracy.

5. Performance results

In this section, we report and analyze the performance of our system with different configurations and compare it with other methods in the literature.

5.1. Extended-MNIST dataset

The first dataset which is used to train the system is EMNIST-Digits [9]. This dataset is about digit recognition and has the same attributes as MNIST [19] but with almost 5 times the number of samples. Table 1 contains the results of the three architectures mentioned in Section 4.1 along with other existing top-rank methods. Figure 5 visualizes the same table. By adding the optical convolution layer to

Table 1: Accuracy, size and number of MAdds operations for the best performing models on EMNIST dataset.

	Method	Accuracy on Digits	Accuracy on Letters	Accuracy on Balanced	# Parameters	MAdds
A	Syntiant (4 Dense)	95.16%	76.54%	70.21%	0.55M	**0.55M**
B	OPIUM[9]	95.90%	85.15%	78.02%	8.32M	8.32M[1]
C	Autoencoder[40]	-	91.27%	-	0.35M	2.3M
-	HM2-BP[17]	-	-	85.57%	0.67M	NA
D	CNN (2 Conv, 2 Dense)[5]	99.46%	93.63%	87.18%	1.2M	13.95M
E	Parallelized CNN[34]	99.62%	-	-	**0.21M**	6.13M
F	NeuroEvolved CNN[2]	99.73%	95.19%	-	2.03M	100.19M
-	EDEN[11]	99.30%	-	88.30%	1.69M	NA
G	TextCaps[16]	**99.79%**	**95.36%**	90.46%	5.87M	253.48M[1]
H	CNN (6 Conv, 2 Dense)[31]	**99.79%**	-	**90.59%**	1.36M	131.31M
I	OptConv+Perc (ours)	98.29%	91.92%	84.68%	0.55M	**0.55M**
J	OptConv+Log+Perc (ours)	99.43%	93.65%	87.69%	0.55M	**0.55M**

[1]Number of MAdds operations computed only for the linear and convolutional part of the method. Extra steps (external classifiers, network capsules, etc.) are not taken into account.

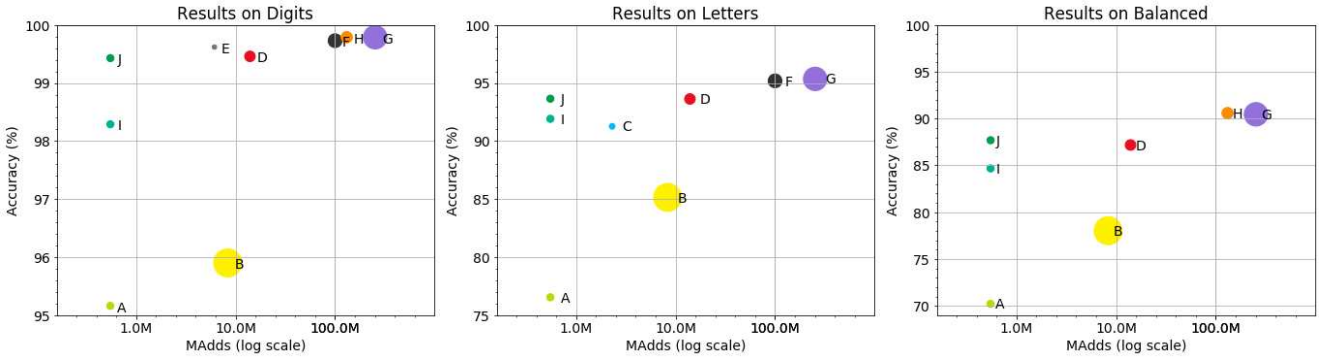


Figure 5: Best viewed on color screen. We show a comparison of accuracy versus number of operations between different methods on the EMNIST-Digits, EMNIST-Letters and EMNIST-Balanced datasets. The alphabetic and color-coding to each method is described in Table 1.

the perceptron network, we can increase the accuracy from 95.16% to 98.29% which is equivalent to decreasing the error rate by 65%. Afterwards, as explained in Section 4.2.1, by continuing the training while replacing the linear image sensor quantizer by a logarithmic one, the accuracy further increases to 99.43%. This results in 42% decrease of the error rate. It is interesting to note that the computational cost (number of multiplication-addition operations) is reduced by a factor of 250 while the accuracy is decreased only by 0.36%. Also, the number of network parameters in the processing unit (the amount of memory required to keep the network) decreases by a factor of 2.5. It should also be noted that having the big convolution in the optical domain has saved 1.25M multiplication-addition operations and 57K parameters.

We then repeated the same experiments for the EMNIST-Letters dataset which contains the hand-written samples of the 26 letters of English alphabet. The accuracy, number of parameters and computational cost of the proposed methods along with approaches are also reported in Table 1. Figure 5 visualizes these methods and shows the global trend and the trade off between accuracy and computation effort (we are unable to report the performance of HM2-BP and EDEN since their computational costs are not reported). We can see that adding the optical convolution layer increases the accuracy from 76.54% to 91.92%, and replacing the linear activation function with the logarithmic one improves it to 93.65%. Overall, while decreasing the computational cost by a factor of 460 and the number of parameters by a factor of 10.6, we only lose 1.71% of the accuracy compared to other state-of-the-art methods.

The EMNIST-Balanced dataset includes the digits and letters in 47 classes (in fact, for some letters like 'O', the capital and small cases are considered as a single class). Table 1 and Figure 5 present the results of different techniques on this dataset. In this experiment, we see that the

Figure 6: Sample images from the 7 classes of the InAirGestures dataset.

performance of the perceptron network alone, the optical convolution plus the perceptron network, and the optical convolution with logarithmic activation added to the perceptron network achieve the accuracy of 70.21%, 84.68% and 87.69%, respectively. Similar to the previous experiments, while we reduce the computational cost by a factor of 240 and the number of parameters by a factor of 2.5, we only lose 2.9% of accuracy compared to the state-of-the art.

5.2. InAirGestures dataset

We performed additional experiments on the InAirGestures dataset [35] which contains 7 different static gestures (6 meaningful gestures and 1 no gesture class). Some sample images are depicted in Figure 6. The training and testing sets contain between 2000 and 2500 samples for each gesture. To process the data, each image was cropped in the center, resulting in 240×240 images. The images were resized to 80×80 and converted to grayscale. For the optical convolution kernel size we used again 240×240 and stride 6 (to obtain a 40×40 input for the perceptron network). For the baseline model, we resize the images to 40×40 and use that as the input to the perceptron network. After training both networks, the baseline model reached 97.16% accuracy on test set while the model with optical convolution reached 99.94% accuracy, which again shows the advantage of using the large kernel.

6. Summary and future works

In this paper, we proposed a novel realization of performing convolutional operations in the optical domain before acquiring the image, as a first layer of a convolutional neural network. The proposed optical front-end uses a high resolution coded aperture, which includes tens of thousands of parameters. As this convolution happens in the optical domain, it has zero cost in terms of energy consumption and also effective zero latency (computational operations). Additionally, there is no limit on the size of the kernel of convolution (in terms of the number of parameters) as long as the pixel size of the kernel is large enough compared to the maximum wavelength of the visible light ($0.7\,\mu\mathrm{m}$).

Based on this idea, we proposed a vision system with less than a milli-watt power consumption (below $800\,\mu\mathrm{W}$ for image sensing and data transfer, and $150\,\mu\mathrm{W}$ for data processing) for low-power and real-time OCR application. We demonstrated the effect of the big optical convolutional layer (kernel size of 240×240) in three variations of OCR applications. We observed that we obtain almost state-of-the-art performance with two orders of magnitude less computational cost. Moreover, having an image sensor which enables reading a non-linear transformation of the pixel values, such as ERGO image sensor [29] that outputs the logarithm of the pixel values, results in gaining better performance without adding any computational load.

The similarity of the proposed approach with the extremely efficient mammalian visual system suggests that there is a strong merit in this approach. Both of these methods record the image in a transformed domain rather than taking raw intensity values [24]. In fact, the idea of recording the scene as a spatial grid of luminosity values, which is how the conventional cameras work, is inefficient in several aspects. That is why in most image processing and computer vision applications the images undergo some transformation (Fourier, wavelets and etc.) for efficient processing. Therefore, the ability to capture the scene in optimal (for each task) transformation domain using optical convolutions, can result in more efficiency of software computation.

Possible extensions to this work include: a) finding the optimal neural network architecture after the large optical convolution layer, b) finding a universal optical kernel which is suitable for different applications, and c) learning binary (0 and 1) optical kernels which result in dramatic simplification of the physical implementation of the mask. Furthermore, using this method to construct efficient vision systems for applications such as image compression, and extracting more information from the scene (e.g. depth estimation) open up other interesting future works. Eventually, studying the possibility of stacking a series of convolutional layers with optical non-linearities in between can result in the full implementation of a convolutional neural network in the optical domain.

Appendix I: Preliminaries on the light-field transformations in an optical system

In this section, we propose an optical system containing one or two lenses and a spatially coded transmission mask whose output is the convolution of the image plane with the mask. We use the light-field model in order to explain how the system is designed and works [23].

A light-field is a real-valued function $L(\mathbf{x}, \mathbf{u})$ on a 3-dimensional vector field that gives the luminosity of light for any position $\mathbf{x}$ in the space and for any direction $\mathbf{u}$. Since the luminosity of light is constant over each ray, this function can have a lower dimensional representation. For a given plane $\mathcal{P}$, we characterize the light field on this plane with $L(\mathbf{x}, \mathbf{u})$ from $\mathbb{R}^2 \times \mathbb{R}^2$ to $\mathbb{R}^+$. Here, $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ indicates the location on this plane, $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$ is such that the 3-dimensional vector $(u_1, u_2, 1) \in \mathbb{R}^3$ indi-

cates the ray direction and $\mathbb{R}^+$ is the set of non-negative real numbers. In fact, $L(\mathbf{x}, \mathbf{u})$ is the luminosity of light at position $\mathbf{x}$ in the direction of $(u_1, u_2, 1)$. Notice that since luminosity of each ray is constant over the whole direction, the function $L(\mathbf{x}, \mathbf{u})$ describes the light-field in the whole 3-dimensional space. Since we are going to study a camera, the planes of interest for us are the ones that are orthogonal to the optical axis. We assume that our optical axis is along Z-axis and the origin of coordinates in the planes of interest is at their intersection with the optical axis.

Having $L(\mathbf{x}, \mathbf{u})$ over the plane $\mathcal{P}$, the light-field at plane $\mathcal{P}'$ parallel to $\mathcal{P}$ with distance d is

$$L'(\mathbf{x}, \mathbf{u}) = L(\mathbf{x} - d\mathbf{u}, \mathbf{u}). \tag{5}$$

Also, due to the rules of lenses, the light-field after a lens with focal length f is

$$L'(\mathbf{x}, \mathbf{u}) = L\Bigg(\left(1 - \frac{d}{f}\right)\mathbf{x} - dd'\left(\frac{1}{d} + \frac{1}{d'} - \frac{1}{f}\right)\mathbf{u}, \frac{1}{f}\mathbf{x} + \left(1 - \frac{d'}{f}\right)\mathbf{u}\Bigg) \tag{6}$$

where d and d' are the distances of the planes $\mathcal{P}$ and $\mathcal{P}'$ from the lens, respectively. Moreover, if $\mathcal{P}$ is the object plane, then we have

$$L(\mathbf{x}, \mathbf{u}) = J(\mathbf{x}) \tag{7}$$

which means that the light-field is only a function of the location.

On the other hand, the imaging process (measuring the pixel values) at the sensor plane $\mathcal{P}'$ can be formulated as

$$I(\mathbf{x}) = \int L'(\mathbf{x}, \mathbf{u}) T(\mathbf{u}) \mathrm{d}\mathbf{u} \tag{8}$$

in which $I(\mathbf{x})$ is the measured value at position $\mathbf{x}$ (pixel value), $L'(\mathbf{x}, \mathbf{u})$ is the light-field at plane $\mathcal{P}'$ and $T(\mathbf{u})$ is the absorbance of the incident ray with direction $\mathbf{u}$. Here we focus on the simplified case of $T(\mathbf{u}) = \frac{1}{2\pi}(1 + \|\mathbf{u}\|_2^2)^{-\frac{3}{2}}$.

The last component that we need to define is a spatially coded transmission mask. A spatially coded transmission mask can be described by $K(\mathbf{x}) : \mathbb{R}^2 \to [0, 1]$. It attenuates the luminosity of the rays that arrive at position $\mathbf{x}$ by a factor of $K(\mathbf{x})$ which is a value between 0 and 1 (the rest is absorbed by the masking material).

Appendix I-A: Proposed architecture to perform convolution in optical domain

Assume the optical setup depicted in Figure 7. Using Equations 5-8, we obtain the following relation between the recorded value on the sensor and the object:

$$I(\mathbf{x}) = \int J(-\alpha\mathbf{x} - \mathbf{u}) K(-\gamma\mathbf{u}) T(\gamma\mathbf{u}) \, \mathrm{d}\mathbf{u} \tag{9}$$

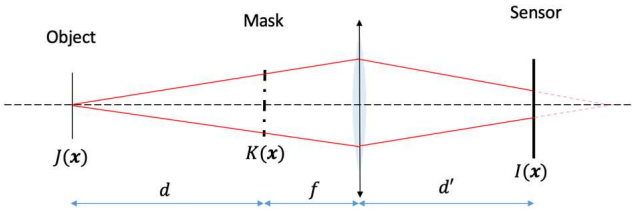


Figure 7: Single lens optical setup for spatial convolution.

in which the $I(\mathbf{x})$ is the recorded value on the sensor at position $\mathbf{x}$, $J(\cdot)$ is the light-field on object plane (which is only a function of location), $K(\cdot)$ is the transmission mask, $\alpha = \frac{d}{f}$, and $\gamma = (1 + \alpha\,(1 - \beta))^{-1}$ where $\beta = \frac{d'}{f}$. Reformulating this equation, we can write

$$I(\mathbf{x}) = (J(\alpha\,\cdot) * \gamma K(\gamma\,\cdot) T(\gamma\,\cdot))\,(-\mathbf{x}) \tag{10}$$

which is a scaled convolution between $J(\cdot)$ and $K(\cdot)T(\cdot)$. Notice that by changing the parameters f, d and d', we can change the scaling factors of the object and the mask.

We can also put a second objective lens to have more control on the parameters of the system. The optical configuration of such system along with the corresponding parameters are in Figure 1 and Equation 2, respectively.

Remark 1 *There are two special cases:*

1. *If*
$$e = f_2 - f_1\left(1 - \frac{d'}{f_1}\right)^{-1}, \tag{11}$$
then the kernel mask magnification is
$$\gamma = \left(\frac{f_2}{f_1}\left(1 - \frac{d'}{f_1}\right)\right)^{-1} \tag{12}$$
that is independent of the object distance d.

2. *If $e = f_2$, then the object magnification is $\alpha = \frac{f_2}{f_1}$ that is independent of the object distance d.*

References

[1] Syntiant® always-on speech & audio recognition processors ndp101. www.syntiant.com/ndp101. 3, 4

[2] Alejandro Baldominos, Yago Saez, and Pedro Isasi. Hybridizing evolutionary computation and deep neural networks: An approach to handwriting recognition using committees and transfer learning. *Complexity*, 2019. 6

[3] Siavash Arjomand Bigdeli, Matthias Zwicker, Paolo Favaro, and Meiguang Jin. Deep mean-shift priors for image restoration. In *Advances in Neural Information Processing Systems*, pages 763–772, 2017. 2

[4] H. John Caulfield, Jason Kinser, and Steven K. Rogers. Optical neural networks. *Proceedings of the IEEE*, 77(10):1573–1583, 1989. 1

[5] Paulo Cavalin and Luiz Oliveira. Confusion matrix-based building of hierarchical classification. In *Iberoamerican Congress on Pattern Recognition*, pages 271–278. Springer, 2018. 6

[6] Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific Reports*, 8, 2018. 1

[7] S. Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE transactions on image processing*, 9(9):1532–1546, 2000. 2

[8] Huaijin G. Chen, Suren Jayasuriya, Jiyue Yang, Judy Stephen, Sriram Sivaramakrishnan, Ashok Veeraraghavan, and Alyosha Molnar. Asp vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 903–912, 2016. 1

[9] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017. 5, 6

[10] Shane Colburn, Yi Chu, Eli Shilzerman, and Arka Majumdar. Optical frontend for a convolutional neural network. *Applied optics*, 58(12):3179–3186, 2019. 1

[11] Emmanuel Dufourq and Bruce A. Bassett. Eden: Evolutionary deep networks for efficient machine learning. In *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, pages 110–115, 2017. 6

[12] Uzi Efron. *Spatial light modulator technology: materials, devices, and applications*, volume 47. CRC press, 1994. 4

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 1

[14] Harel Haim, Shay Elmalem, Raja Giryes, Alex M. Bronstein, and Emanuel Marom. Depth estimation from a single image using deep learned phase coded mask. *IEEE Transactions on Computational Imaging*, 4(3):298–310, 2018. 1

[15] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, and William J. Dally. Eie: efficient inference engine on compressed deep neural network. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pages 243–254, 2016. 1

[16] Vinoj Jayasundara, Sandaru Jayasekara, Hirunima Jayasekara, Jathushan Rajasegaran, Suranga Seneviratne, and Ranga Rodrigo. Textcaps: Handwritten character recognition with very small datasets. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 254–262, 2019. 6

[17] Yingyezhe Jin, Wenrui Zhang, and Peng Li. Hybrid macro/micro level backpropagation for training deep spiking neural networks. In *Advances in Neural Information Processing Systems*, pages 7005–7015, 2018. 6

[18] Sanjeev J. Koppal, Ioannis Gkioulekas, Travis Young, Hyunsung Park, Kenneth B. Crozier, Geoffrey L. Barrows, and Todd Zickler. Toward wide-angle microvision sensors. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2982–2996, 2013. 1

[19] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 3, 5

[20] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10):959–971, 1996. 2

[21] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70–es, 2007. 1

[22] Xing Lin, Yair Rivenson, Nezih T. Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan. All-optical machine learning using diffractive deep neural networks. *Science*, 361(6406):1004–1008, 2018. 1

[23] Ren Ng et al. *Digital light field photography*. Stanford University, 2006. 7

[24] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996. 2, 7

[25] Pedram Pad, Kasra Alishahi, and Michael Unser. Optimized wavelet denoising for self-similar α-stable processes. *IEEE Transactions on Information Theory*, 63(9):5529–5543, 2017. 2

[26] Pedram Pad, Farnood Salehi, Elisa Celis, Patrick Thiran, and Michael Unser. Dictionary learning based on sparse distribution tomography. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2731–2740, 2017. 2

[27] Pedram Pad, Virginie Uhlmann, and Michael Unser. Maximally localized radial profiles for tight steerable wavelet frames. *IEEE Transactions on Image Processing*, 25(5):2275–2287, 2016. 2

[28] Pedram Pad and Michael Unser. Optimality of operator-like wavelets for representing sparse AR(1) processes. *IEEE Transactions on Signal Processing*, 63(18):4827–4837, 2015. 2

[29] Pierre-Francois Ruedi, Pascal Heim, Steve Gyger, Francois Kaess, Claude Arm, Ricardo Caseiro, Jean-Luc Nagel, and Silvio Todeschini. An soc combining a 132db qvga pixel array and a 32b dsp/mcu processor for vision applications. In *2009 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*, pages 46–47, 2009. 3, 4, 7

[30] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 1

[31] Ashadullah Shawon, Md Jamil-Ur Rahman, Firoz Mahmud, and M. M. Arefin Zaman. Bangla handwritten digit recognition using deep cnn for large and unbiased dataset. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6, 2018. 6

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[33] Madhusudan Singh, Hanna M Haverinen, Parul Dhagat, and Ghassan E. Jabbour. Inkjet printing process and its applications. *Advanced materials*, 22(6):673–685, 2010. 4

[34] Srishti Singh, Amrit Paul, and M. Arun. Parallelization of digit recognition system using deep convolutional neural network on cuda. In *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, pages 379–383, 2017. 6

[35] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 319–329, 2014. 7

[36] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015. 1

[37] Kelvin Wagner and Demetri Psaltis. Optical neural networks: an introduction by the feature editors. *Appl. Opt.*, 32(8):1261–1263, Mar 1993. 1

[38] Zihao W. Wang, Vibhav Vineet, Francesco Pittaluga, Sudipta N. Sinha, Oliver Cossairt, and Sing Bing Kang. Privacy-preserving action recognition using coded aperture videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, 2019. 1

[39] N. J. Wilkinson, M. A. A. Smith, R. W. Kay, and R. A. Harris. A review of aerosol-jet printing – a non-traditional hybrid process for micro-manufacturing. *The International Journal of Advanced Manufacturing Technology*, pages 1–21, 2019. 4

[40] Rey Wiyatno and Jeff Orchard. Style memory: Making a classifier network generative. In *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*, pages 16–21, 2018. 6