



(51) International Patent Classification:

G06N 3/04 (2006.01)

G06N 3/067 (2006.01)

(21) International Application Number:

PCT/US2019/060935

(22) International Filing Date:

12 November 2019 (12.11.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

62/758,735 12 November 2018 (12.11.2018) US

62/798,267 29 January 2019 (29.01.2019) US

(71) Applicant: **MASSACHUSETTS INSTITUTE OF TECHNOLOGY** [US/US]; 77 Massachusetts Avenue, Cambridge, MA 02139 (US).

(72) Inventors; and

(71) Applicants (for US only): **HAMERLY, Ryan** [US/US]; 332 Walden St, Cambridge, MA 02138 (US). **ENGLUND,**

Dirk Robert [US/US]; 116 Pleasant Street #3, Brookline, MA 02446 (US).

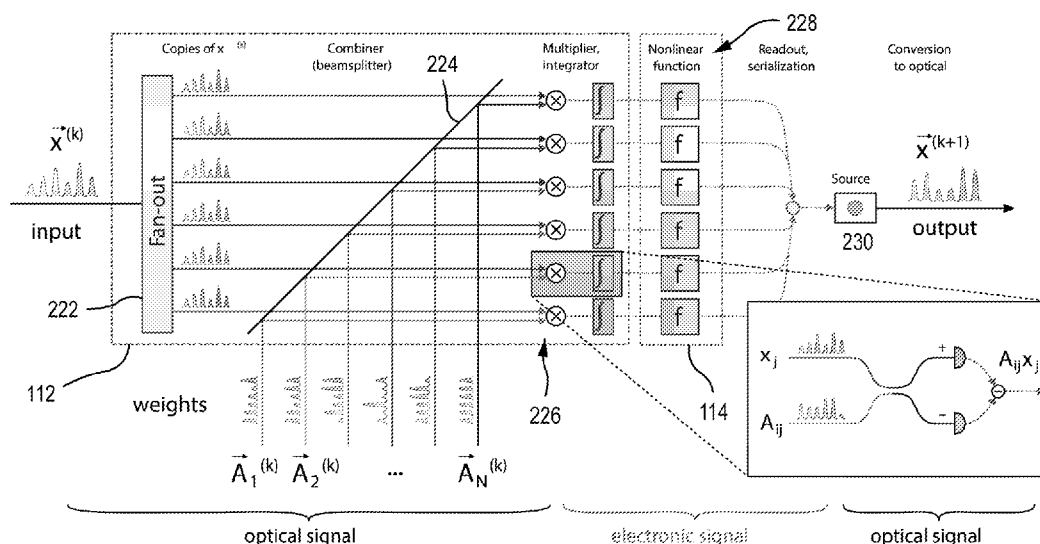
(74) Agent: **COLICE, Christopher Max** et al.; Smith Baluch LLP, 376 Boylston St., Suite 401, Boston, MA 02116 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

(54) Title: LARGE-SCALE ARTIFICIAL NEURAL-NETWORK ACCELERATORS BASED ON COHERENT DETECTION AND OPTICAL DATA FAN-OUT

FIG. 2



(57) Abstract: Deep learning performance is limited by computing power, which is in turn limited by energy consumption. Optics can make neural networks faster and more efficient, but current schemes suffer from limited connectivity and the large footprint of low-loss nanophotonic devices. Our optical neural network architecture addresses these problems using homodyne detection and optical data fan-out. It is scalable to large networks without sacrificing speed or consuming too much energy. It can perform inference and training and work with both fully connected and convolutional neural-network layers. In our architecture, each neural network layer operates on inputs and weights encoded onto optical pulse amplitudes. A homodyne detector computes the vector product of the inputs and weights. The nonlinear activation function is performed electronically on the output of this linear homodyne detection step. Optical modulators combine the outputs from the nonlinear activation function and encode them onto optical pulses input into the next layer.

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

Large-Scale Artificial Neural-Network Accelerators Based on Coherent Detection and Optical Data Fan-Out

CROSS-REFERENCE TO RELATED APPLICATION(S)

[001] This application claims priority, under 35 U.S.C. § 119(e), to U.S. Application No. 62/798,267, filed January 29, 2019, and to U.S. Application No. 62/758,735, filed November 12, 2018, each of which is incorporated herein by reference in its entirety.

GOVERNMENT SUPPORT STATEMENT

[002] This invention was made with Government support under Grant No. FA9550-16-1-0391 awarded by the Air Force Office of Scientific Research (AFOSR) and under Grant No. W911NF-18-2-0048 awarded by the Army Research Office. The Government has certain rights in the invention.

BACKGROUND

[003] In recent years, deep neural networks have successfully tackled a wide range of problems including image analysis, natural language processing, game playing, physical chemistry, and medicine. The recent interest in deep neural networks is driven by (1) the availability of large training datasets and (2) substantial growth in computing power and the ability to train networks on graphics processing units (GPUs). But moving to more complex problems and higher network accuracies requires larger and deeper neural networks, which in turn require even more computing power. This motivates the development of special-purpose hardware optimized to perform neural-network inference and training.

[004] To outperform a GPU, a special-purpose neural-network accelerator should have significantly lower energy consumption than a GPU since the performance of modern GPUs is limited by on-chip power, which in turn is limited by the heatsink. In addition, the neural-network accelerator should be fast, programmable, scalable to many neurons, compact, and compatible with training as well as inference.

[005] An application-specific integrated circuits (ASIC) is one candidate for a special-purpose neural-network accelerator. A state-of-the-art ASIC can reduce the energy per multiply-and-accumulate (MAC) operation from 20 pJ/MAC for modern GPUs to around 1 pJ/MAC. However, ASICs are based on CMOS technology and therefore suffer from the interconnect problem—even in highly optimized architectures where data is stored in register files close to the logic units, most of the energy consumption comes from data movement, not logic. Analog crossbar arrays based on CMOS gates or memristors promise better performance but are limited in size and suffer from crosstalk and calibration issues.

[006] By contrast, photonic neural-network accelerators reduce both the logic and data-movement energy consumption by performing (the linear part of) each neural-network layer in a passive, linear optical circuit. In this approach, the linear step is performed at high speed with no energy consumption beyond transmitter and receiver energy consumption. Optical neural networks based on free-space diffraction have been reported but require spatial light modulators or 3D-printed diffractive elements and are therefore not rapidly programmable. Nanophotonic circuits are a promising alternative, but the large footprint of directional couplers and phase modulators precludes scaling to large (e.g., $N \geq 1000$) numbers of neurons. To date, the goal of a large-scale, rapidly reprogrammable photonic neural network remains unrealized.

SUMMARY

[007] Progress in deep learning has led to a resource crunch where performance is limited by computing power, which is in turn limited by energy consumption. Optics can increase the speed and reduce the energy consumption of neural networks, but current optical architectures suffer from limited connectivity and the large footprint of low-loss nanophotonic devices. The optical architectures presented here use homodyne detection and optical data fan-out to circumvent these limits. These optical architectures are scalable to large (e.g., millions of neurons) networks without sacrificing speed (e.g., GHz) or energy consumption (e.g., sub-fJ/operation).

[008] One of these optical architectures is an optical neural network that includes at least one coherent light source, first and second optical fan-out elements, a two-dimensional array of homodyne receivers in optical communication with the first and second optical fan-out elements, electronic circuitry operably coupled to the two-dimensional array of homodyne receivers, and a

light source operably coupled to the electronic circuitry. In operation, the coherent light source produces N optical weight signals, which are fanned out into M copies by the first optical fan-out element. (Here, M and N are positive integers, where N can be about 1,000 to about 10,000,000.) The second optical fan-out element create N copies of M optical input signals. The homodyne receivers produce electronic signals representing interference of each of the N copies of the M optical input signals with a corresponding copy of the M copies of the N optical weight signals. The electronic circuitry applies a nonlinear function to the electronic signals. And the light source emits an optical output signal representing a matrix multiplication of the M optical input signals with the N optical weight signals. For example, the optical input signal may encode a vector length N and each corresponding optical weight signal represents a row of a weight matrix that are multiplied together.

[009] In some cases, the first and second optical fan-out elements, two-dimensional array of homodyne receivers, electronic circuitry, and light source form a convolutional layer in a series of layers in the optical neural network. The series of layers may include additional convolutional layers and at least one fully connected layer. There may also be another series of layers, in optical communication with the coherent light source, to compute a matrix multiplication of the N optical weight signals with another M optical input signals.

[0010] The N copies of the M optical input signals may propagate in free space between the second fan-out element and the array of homodyne receivers.

[0011] Each homodyne receiver in the array of homodyne receiver can include a two-port beam splitter and a differential detector in optical communication with the two-port beam splitter. The two-port beam splitter interferes the copy of the optical input signal and the corresponding optical weight signal. And the differential detector detects light emitted by outputs of the two-port beam splitter.

[0012] The electronic circuitry can include an array of analog-to-digital converters (ADCs), each of which is operably coupled to a corresponding homodyne receiver in the array of homodyne receivers; digital logic operably coupled to the array of ADCs; and an array of digital-to-analog converters (DACs) operably coupled to the digital logic and the light source. Each ADC digitizes the electronic signal emitted by the corresponding homodyne receiver. The digital logic applies

the nonlinear function to the electronic signals from the array of ADCs. And the DACs convert the electronic signals into analog signals for modulating the light source.

[0013] The coherent light source may include a pulsed laser that emits an optical pulse train, in which case a beam splitter may split the optical pulse train into an array of optical pulse trains. An array of optical modulators, in optical communication with the beam splitter, may modulate each optical pulse train in the array of pulse trains with weights representing a corresponding row in a weight matrix.

[0014] The optical neural network may also include a beam splitter, in optical communication with the first optical fan-out element and the second optical fan-out element, to combine each of the N copies of the M optical input signals with the corresponding copy of the M copies of the N optical weight signals.

[0015] Other optical processors may implement a method for fanning out data in a digital optical neural network. This method may include, for each image in a set of images that represents an input to a layer in the digital optical neural network, breaking the image into a set of patches, each of which comprises $K_x \times K_y$ pixels. For each of these patches, the processor spatially convolving the patch with a box function. The processor also spatially convolving a corresponding kernel with a size of $K_x \times K_y$ with a lattice function having a horizontal lattice spacing of K_x and a vertical lattice spacing of K_y . The processor then images the patch and the corresponding kernel onto a detector array.

[0016] Another example is a convolutional layer for a coherent optical neural network. This convolutional layer includes an image-plane transceiver array, a weight server transmitter array, a beam splitter in optical communication with the image-plane transceiver array and the weight server transmitter array, and a Fourier-plane transceiver array in a Fourier plane of the image-plane transceiver array and in optical communication with the beam splitter. In operation, the image-plane transceiver array emits an array of input signals and the weight server transmitter array emits an array of weight signals. The beam splitter combines the array of input signals with the array of weight signals. And the Fourier-plane transceiver array detects a homodyne product of a spatial Fourier transform of the array of input signals and a spatial Fourier transform of the array of weight signals.

[0017] The Fourier-plane transceiver array can emit an array of product signals representing the homodyne product of the spatial Fourier transform of the array of input signals and the spatial Fourier transform of the array of weight signals. In this case, the image-plane transceiver array can coherently detect a spatial Fourier transform of the array of product signals. Each transceiver in the Fourier-plane transceiver array can detect an in-phase component and a quadrature component of the product of the spatial Fourier transform of the array of input signals and the spatial Fourier transform of the array of weight signals. And each transceiver in the Fourier-plane transceiver array may include at least one detector element per output channel of the convolutional layer.

[0018] All combinations of the foregoing concepts and additional concepts discussed in greater detail below (provided such concepts are not mutually inconsistent) are part of the inventive subject matter disclosed herein. In particular, all combinations of claimed subject matter appearing at the end of this disclosure are part of the inventive subject matter disclosed herein. The terminology used herein that also may appear in any disclosure incorporated by reference should be accorded a meaning most consistent with the particular concepts disclosed herein.

BRIEF DESCRIPTIONS OF THE DRAWINGS

[0019] The skilled artisan will understand that the drawings primarily are for illustrative purposes and are not intended to limit the scope of the inventive subject matter described herein. The drawings are not necessarily to scale; in some instances, various aspects of the inventive subject matter disclosed herein may be shown exaggerated or enlarged in the drawings to facilitate an understanding of different features. In the drawings, like reference characters generally refer to like features (e.g., functionally similar and/or structurally similar elements).

[0020] FIG. 1 illustrates the dataflow in a typical deep neural network, which includes a sequence of layers containing linear matrix-vector products and nonlinear activation functions.

[0021] FIG. 2 illustrates a homodyne optical neural network accelerator with temporally encoded weights. As shown, this accelerator implements a single layer in the neural network of FIG. 2A.

[0022] FIG. 3A illustrates a three-layer neural network with full connectivity.

[0023] FIG. 3B illustrates classification of an MNIST image classified by the three-layer neural network in FIG. 3A (size $784 \rightarrow 1000 \rightarrow 1000 \rightarrow 10$).

[0024] FIG. 3C is a plot of error rate as a function of photons per MAC n_{mac} (equivalently energy $E_{\text{mac}} = (hc/\lambda)n_{\text{mac}}$, here $\lambda = 1.55\mu\text{m}$) for different sized versions of the three-layer neural network of FIG. 3A.

[0025] FIG. 4A is a plot of error rate as a function of number of MACs for different fully-connected neural networks trained on the MNIST database.

[0026] FIG. 4B is a plot of error rate as a function of total number of photons, for the same networks as in FIG. 4A.

[0027] FIG. 4C is a plot of contributions to the neural network energy budget, with standard quantum limit (SQL) dots corresponding to the minimum E_{mac} required to make the error rate $p_{\text{err}}(E_{\text{mac}}) < 1.5p_{\text{err}}(\infty)$ (error bars correspond to $p_{\text{err}}(E_{\text{mac}}) = [1.2, 2.0]p_{\text{err}}(\infty)$). $E_{\text{mac}} = n_{\text{mac}}(hc/\lambda)$, $\lambda = 1.55\mu\text{m}$. This SQL is calculated using the error rates in FIGS. 3B and 3C.

[0028] FIG. 5A illustrates matrix multiplication with a 2D detector array, two 1D transmitter arrays, and optical fan-out.

[0029] FIG. 5B is a schematic diagram of transmitter array suitable for performing matrix multiplication as in FIG. 5A.

[0030] FIG. 5C illustrates matrix operations for inference, training and back-propagation in a deep neural network.

[0031] FIG. 5D illustrates patching technique to recast a convolution ($K_x = K_y = 3, s_x = s_y = 2$ shown) as a matrix-matrix multiplication.

[0032] FIG. 6A is a schematic drawing of AlexNet, which includes 5 convolutional layers and 3 fully-connected layers (the pooling and normalization steps not shown).

[0033] FIG. 6B is a plot of error rate for pre-trained AlexNet as a function of n_{mac} (dashed lines show the effect of noise in a single layer, while the solid line shows the performance of the actual machine where all layers have noise).

[0034] FIG. 7A shows the data fan-out in a convolutional layer: specifically how outputs $y_{ij;m}$ (right) depend on a particular input $x_{kl;n}$ (upper left) and a particular weight $K_{kl;mn}$ (lower left).

[0035] FIG. 7B illustrates a convolution as a weighted fan-in to a single output pixel y_{ij} .

[0036] FIG. 7C is a fan-out representation of a convolution.

[0037] FIG. 8 illustrates a patch method for input fan-out in a digital optical convolutional neural network.

[0038] FIG. 9A shows conversion of a point source to a plane wave in the Fourier plane, whose amplitude and/or phase are modified with a spatial light modulator (SLM), to produce a signal that is Fourier-transformed to form a hologram.

[0039] FIG. 9B illustrates multiple sources producing multiple non-overlapping holograms, correctly implementing an optical convolution.

[0040] FIG. 10 illustrates the shift method for input and kernel fan-out in a digital optical convolutional neural network.

[0041] FIG. 11A is a bar chart of AlexNet energy consumption broken down into logic (darker bar segments) and memory access (lighter bar segments).

[0042] FIG. 11B is a circuit diagram of an analog weighted accumulator restricted to positive signals/weights.

[0043] FIG. 11C is a circuit diagram of a weighted accumulator for both positive and negative signals/weights. Result = (Out₊) – (Out₋).

[0044] FIG. 12A illustrates a process for performing a convolution using three Fourier transforms and a multiplication.

[0045] FIG. 12B illustrates Fourier transformation of weights and feature maps and detection of the product $\tilde{K}_{kl}^* \tilde{x}_{kl}$ in the Fourier plane as part of an optical implementation of the convolution process in FIG. 12A.

[0046] FIG. 12C illustrates Fourier transformation and coherent detection of the product $(\tilde{K}_{kl}^* \tilde{x}_{kl})^*$ in the image plane, producing the convolution $(\mathcal{F}^{-1}[\tilde{K}^* \tilde{x}])^*$ as part of an optical implementation of the convolution process in FIG. 13A.

[0047] FIG. 13 illustrates a transceiver chip suitable for use in the Fourier plane in the optical implementation of the convolution process in FIGS. 12B and 12C.

[0048] FIG. 14A illustrates C-ordering for serializing channel data with $C = 5$ channel input and $C' = 4$ channel output layer (spatial dimensions (i, j) are suppressed and only channel dimensions (m, n) are shown for clarity).

[0049] FIG. 14B illustrates Fortran ordering for serializing channel data with $C = 5$ channel input and $C' = 4$ channel output layer (spatial dimensions (i, j) are suppressed and only channel dimensions (m, n) are shown for clarity).

[0050] FIG. 15A illustrates encoding channels in frequency space using wavelength-division multiplexing (WDM).

[0051] FIG. 15B illustrates an implementation of a many-channel WDM using rings and delay-Mach-Zehnder modulator (MZM) filters.

[0052] FIG. 16 shows a confocal setup to produce an optical Fourier transform ($x' > 0$ corresponds to below the optical axis).

[0053] FIG. 17 illustrates an optical architecture for performing a multi-channel optical Fourier transform, with subpixels ($C' = 4$) in the Fourier plane and additional emitters in the weight server.

[0054] FIG. 18 illustrates a Row-Column method for computing a convolution.

[0055] FIG. 19 illustrates a Displaced Image Method for computing a convolution.

[0056] FIG. 20 illustrates an optical system for performing the Row-Column method of FIG. 18 and the Displaced Image Method of FIG. 19.

DETAILED DESCRIPTION

[0057] The Detailed Description of this specification is divided into different sections. Section 1 discloses additional details and examples of optical neural network accelerators, including a coherent optical matrix-vector multiplier (FIGS. 2A and 2B) and the extension to matrix-matrix (batched) operations (FIGS. 5A–5D). Section 2 discloses a related “digital” optical neural network, where the computation is performed in digital logic while the optics are used for data movement and fan-out. This is a well-adapted application of optical interconnects to deep neural networks. Section 3 discloses optical convolutional neural networks that combine the concepts in Sec. 1 and

classical Fourier optics. Section 4 discloses alternative techniques for optical convolutional neural networks that use optical data fan-out and may be less complex than the Fourier-optical setup.

1. Large-Scale Optical Neural Networks based on Coherent Detection

[0058] A system that performs the matrix-vector multiplication for a neural network optically using coherent (homodyne) detection can be fast, low-power, compact, and readily scalable to large ($N \gtrsim 10^6$) numbers of neurons. In contrast to other systems, it encodes both the inputs and weights in optical signals, allowing the weights to be changed on the fly at high speed. The ultra-high data bandwidth of multimode free-space optics and the high pixel density of CMOS photodetectors allows this system to scale to far more neurons than can be supported in nanophotonics. The optical energy consumption is subject to a fundamental standard quantum limit (SQL) arising from the effects of shot noise in the homodyne detectors, which lead to classification errors. Simulations based on MNIST neural networks empirically show the SQL to be 50–100 zJ/MAC. Using realistic laser, modulator, and detector energies, performance at the sub-fJ/MAC level should be possible with present technology. The optical system can be used for both fully connected and convolutional layers. Finally, backpropagation is straightforward to implement in this optical system, allowing both inference and training to be performed in the same optical device.

Coherent Matrix Multiplier

[0059] FIG. 1 illustrates a deep neural network 100. The neural network is a sequence of K layers 110-1 through 110- K , each including a matrix multiplication element 112 that performs the matrix multiplication $\vec{x} \rightarrow A\vec{x}$ and an element-wise nonlinearity element 114 that executes a corresponding neuron activation function $x_i \rightarrow f(x_i)$. The input into the $(k + 1)^{\text{th}}$ layer 110- $k+1$ is related to the k^{th} layer input 110- k by:

$$x_i^{(k+1)} = f(\sum_j A_{ij}^{(k)} x_j^{(k)}) \quad (1)$$

This optical processor 100 can work with both fully-connected and convolutional neural-network layers.

[0060] FIG. 2 illustrates special-purpose optical processor, also called an optical neural network accelerator, for artificial neural networks that can perform both (a) inference on pre-trained neural networks and (b) neural-network training. In this case, the accelerator is an optical implementation

of a single layer 110 in the deep neural network 100 of FIG. 1 using coherent optical matrix multiplication. Many of these layers 110 can be pipelined together, e.g., as in FIG. 1, to create a single optical neural network with multiple layers. Alternatively, a single unit in FIG. 1 can be used to implement the entire deep neural network 100 with K layers. In this alternative, the layer's outputs are fed back into the input at each of K time steps via a switch (not shown), with one time step per layer in the neural network. The weights are also updated at each time step, so that the input to the layer 110 at the $(k + 1)^{\text{th}}$ time step obeys the relationship given in Eq. (1). Actuating the switch at the output of the layer 210 directs the output at the K^{th} time step to the output of the neural network instead of to the input of the layer 110.

[0061] Referring again to FIG. 2, inputs $\vec{x}$ and weights $\vec{A}_i$ are encoded onto optical pulse amplitudes $\vec{x}$ (while time encoding into pulses is shown, frequency encoding or a time-frequency hybrid is also acceptable). An optical fan-out element 222, such as a cylindrical lens or a $1 \times k$ beam splitter, splits the inputs $\vec{x}$ into copies, each of which is combined with a corresponding weight $\vec{A}_i$ with a beam splitter 224. The beam splitter 224 transmits the copies of the input signal to an array of differential homodyne detectors 226, which detect optical-domain interference between the copies of the input signal $\vec{x}$ and corresponding weight signals $\vec{A}_i$ reflected by the beam splitter 224. As explained below, each weight signal represents one row of a weight matrix. The homodyne detectors 226 produce an electrical domain signal representing the vector product between the input $\vec{x}$ and the weight matrix rows $\vec{A}_i$. This implements the linear step 114 for the layer.

[0062] Circuitry 228 coupled to the differential homodyne detectors 226 applies a nonlinear function (e.g., a Sigmoid function) in the electrical domain to each detector output, which are read out and used to serially modulate a light source 230, such as a laser, that emits a coherent optical carrier modulated with the layer output. This circuitry 228 may include an array of analog-to-digital converters (ADCs), an array of digital logic elements, and an array of digital-to-analog converters (DACs). Each ADC is coupled to a corresponding homodyne detector 226 and converts the analog detector output into a digital signal. A corresponding digital logic element applies the nonlinear function to the digital signal, which is then converted back into the analog domain by a corresponding DAC. The resulting analog signal modulates the light source 230 in series with the analog signals from the other DACs in the circuitry 228. This modulated optical signal emitted by

the light source 230 becomes the input to the next layer 110 in the optical neural network 100. Alternatively, the electrical-domain outputs from the nonlinear step can be combined and encoded onto the optical domain using optical modulators, producing a pulse sequence that is combined with a fan-in element, such as another beam splitter, fed into the next layer 110 of the deep neural network 100.

[0063] The light source 230 in each layer 110 emits light (optical output signals) that is coherent with the light used to generate the optical weights for the subsequent layer 110. More specifically, the light source 230 in layer 110-1 emits light that is coherent with the light used to generate the optical weights in layer 110-2. This allows the weights and optical input signals for a given layer 110 to interfere at the homodyne detectors 226 in that layer 110. This can be accomplished by using a single pulsed laser to generate light that is distributed and modulated to provide weights and optical input/output signals for all of the layers (that is, one laser split $2K$ ways by a $1 \times K$ beam splitter); by using one laser per layer to generate the optical weights and optical input signals for that layer (that is, K lasers, each split two ways, for K layers 110); or by using pairs or arrays of lasers that are coherent with each other.

[0064] For a given layer 110, let N and N' be the number of input and output neurons, respectively. Input (output) data are encoded temporally as N (N') pulses on a single channel as shown in FIG. 2. This encoding, reminiscent of the Coherent Ising Machine, contrasts with other approaches used for neural networks, which encode inputs in separate spatial channels. As there are NN' weights for an $N' \times N$ fully-connected matrix, the weights enter on N' separate channels, each carrying a single matrix row encoded in time. The fan-out element 222 optically fans out the input data to an array of N' detectors 226, so that each detector 226 calculates the homodyne product between the two signals (inset). As long as both signals are driven from the same coherent source and the path-length difference is less than the coherence length, the charge Q_i accumulated by homodyne receiver i is:

$$Q_i = \frac{2\eta e}{\hbar\omega} \int \text{Re}[E^{(\text{in})}(t)^* E_i^{(\text{wt})}(t)] dt \propto \sum_j A_{ij} x_j \quad (2)$$

[0065] Note that, due to properties of the Fourier transform, it is also possible to encode the input (output) data in the frequency domain (e.g., in amplitudes of the teeth of a frequency comb) or in a hybrid fashion (e.g., a pulse train combined with wavelength multiplexing).

[0066] Here $E^{(\text{in})}(t)$ and $E_i^{(\text{wt})}(t)$ are the input and weight fields for receiver i , which are taken to be sequences of pulses with amplitudes proportional to x_j and A_{ij} , respectively ($x_j, A_{ij} \in \mathbb{R}$). Thus, each receiver 226 performs a vector-vector product between $\vec{x}$ and a column $\vec{A}_i$ of the weight matrix; taken together, the N' electronic outputs give the matrix-vector product $A\vec{x}$. Fields are normalized so that power is given by $P(t) = |E(t)|^2$, and η is the detector efficiency. A serializer 228 reads out these values one by one, applies the nonlinear function $f(\cdot)$ in the electrical domain, and outputs the result to a modulated source 230 to produce the next layer's inputs.

[0067] The inset of FIG. 2 shows the homodyne detector 226 in greater detail. The homodyne detector 226 includes a 2×2 beam splitter, with one input port for the input signal, one input port for the weight signal, and one photodiode 234 per output port. The homodyne detector 226 combines the advantages of optics and electronics: it can process data encoded at extremely high speeds, limited only by the bandwidth of the beam splitter 232 ($\gtrsim \text{THz}$) and the (optical) bandwidth of the photodiodes 234 (typically $\gtrsim 100 \text{ nm}$, or $\gtrsim 10 \text{ THz}$). The electrical bandwidth can be much slower, since only the integrated charge is measured. Finally, the optical neural network avoids the need for nonlinear optics that is a major stumbling block in all-optical logic: since the output is electrical, the dot product $A_{ij}x_j$ can be computed at extremely low power (aJ/MAC) using standard non-resonant components (photodiodes 234) that are CMOS-compatible and scalable to arrays of millions.

[0068] Coherent detection greatly simplifies the setup compared to alternative approaches. With a given set of weight inputs, the layer 110 in FIG. 2B takes N input pulses and N' detectors to perform a matrix-vector operation with NN' MACs, performing an operation that should scale quadratically with size using only linear resources. This is in contrast to electrical approaches that require quadratic resources (NN' floating-point operations total). The (optical) energy consumption of nanophotonic systems also scales linearly for the same operation; however, a nanophotonic circuit is much more complex, requiring a circuit with $O(NN')$ phase shifters, which becomes very challenging to scale beyond several hundred channels and may be sensitive to propagation of fabrication errors.

[0069] The optical neural network accelerator in FIG. 2 does consume energy to generate the weights, which imposes an energy cost that scales quadratically. However, in many cases (particularly in data centers) neural networks are run simultaneously over large batches of data, so

with appropriate optical fan-out, the cost of the weights can be amortized over many clients. Put another way, running the neural network on data with batch size B , the optical neural network computes a matrix-matrix product $Y_{N' \times B} = A_{N' \times N} X_{N \times B}$, which involves $N'NB$ MACs, with an energy cost that should scale as $O(N'N) + O(N'B) + O(NB)$ rather than $O(N'NB)$.

[0070] Compared to all-electronic neural network processors, such as CMOS integrated circuits, the optical neural network accelerator shown in FIG. 1 consumes much less (e.g., $\geq 100 \times$ less) energy. Currently, artificial neural networks are run on CMOS integrated circuits, such as CPUs, GPUs and TPUs. The computing power of these circuits is limited by their energy consumption, with fundamental physics-based rules related to wire-charging capacitance that set lower limits to this energy consumption. The optical neural network accelerator in FIG. 1 performs connections optically and does not have the same lower limits on energy consumption as CMOS integrated circuits. In addition, the optical neural network accelerator in FIG. 1 can be scaled to much larger neural networks (e.g., $\geq 10^6$ neurons per layer) than nanophotonic optical machine-learning approaches, which are usually limited by the chip areas of their photonic components.

[0071] The optical neural network accelerator in FIG. 1 can be used for a range of commercial applications, including cloud computing, data centers, and supercomputers, since these applications heavily utilize neural networks and suffer from high energy consumption and high cooling costs. Energy-limited embedded computing (e.g. image/video processing in drones, security cameras, phones) may also be possible. Machine learning is a huge industry. Many large and small companies (e.g., Amazon, Google, FloydHub, and Paperspace) provide cloud-computing GPU services specifically optimized for running deep neural networks. The development of the Google TPU (an ASIC specifically designed for neural-network inference and training) underscores the significant industry investment in developing special-purpose hardware for machine learning.

Deep Learning at the Standard Quantum Limit

[0072] As energy consumption is a primary concern in neuromorphic and computing hardware generally, an optical approach should outperform electronics by a large factor to justify investment in a new technology. In addition, optical systems should show great potential for improvement, e.g., by many orders of magnitude, to allow continued scaling beyond the physical limits of Moore's Law. Thus, it is worth investigating (1) the fundamental, physical limits to the energy

consumption of an optical neural network and (2) the energy consumption of a practical, near-term optical neural network using existing technology.

[0073] Without being bound by any particular theory, the fundamental energy consumption limit for an optical neural network stems from quantum-limited noise. In an electrical signal, energy is quantized at a level $E_{el} = h/\tau_{el}$, where $\tau_{el} \sim 10^{-10}$ s is the signal duration. Optical energy is quantized at a level $E_{opt} = h/\tau_{opt}$, where $\tau_{opt} \equiv c/\lambda \sim (2-5) \times 10^{-15}$ s, which is 10^4-10^5 times higher. As a result, $E_{opt} \gg kT \gg E_{el}$ and electrical signals can be treated in a classical limit governed by thermal noise, while optical signals operate in a zero-temperature quantum limit where vacuum fluctuations dominate. These fluctuations are read out on the photodetectors 234, where the photoelectric effect produces a Poisson-distributed photocurrent. While the photocurrents are subtracted in homodyne detection, the fluctuations add in quadrature, and Eq. (1) is replaced by:

$$x_i^{(k+1)} = f(\sum_j A_{ij}^{(k)} x_j^{(k)} + w_i^{(k)} \frac{\|A^{(k)}\| \|x^{(k)}\|}{\sqrt{N^2 N'}} \frac{\sqrt{N}}{\sqrt{n_{mac}}}) \quad (3)$$

Here the $w_i^{(k)} \sim N(0,1)$ are Gaussian random variables, $\|\cdot\|$ is the L_2 norm, and n_{mac} is the number of photons per MAC, related to the total energy consumption of the layer by $n_{tot} = NN'n_{mac}$. The noise term in Eq. (3) scales as $n_{mac}^{-1/2}$, and therefore the signal-to-noise ratio (SNR) of each layer 220 should scale as $SNR \propto n_{mac}$. Since noise adversely effects the network's performance, the energy minimum should correspond to the value of n_{mac} at which the noise becomes significant.

[0074] To quantify this statement, we perform benchmark simulations using two versions of the three-layer, fully connected neural network shown in FIG. 3A. In the first version, the size of the inputs, two inner layers, and outputs scales as $(784 \rightarrow 100 \rightarrow 100 \rightarrow 10)$. In the second version, the size of the inputs, two inner layers, and outputs scales as $(784 \rightarrow 1000 \rightarrow 1000 \rightarrow 10)$. These networks are trained on the MNIST dataset as shown in FIG. 3B. While MNIST digit classification is a relatively easy task, the intuition developed here should generalize to more challenging problems. In the absence of noise, the networks classify images with high accuracy, as shown in FIG. 3B.

[0075] FIG. 3C shows the error rates for both networks as a function of photons per MAC n_{mac} (equivalently energy $E_{mac} = (hc/\lambda)n_{mac}$, here $\lambda = 1.55\mu\text{m}$). The plots in FIG. 3B indicate that

the error rate is a monotonically decreasing function of n_{mac} . The two asymptotic limits correspond to the noiseless case ($n_{\text{mac}} \rightarrow \infty$, which returns the network's canonical accuracy), and the noise-dominated case ($n_{\text{mac}} \rightarrow 0$, where the network is making a random guess). The cutoff point, which is loosely defined as the lowest possible energy at which the network returns close to its canonical accuracy, and is around 0.5–1 aJ (5–10 photons) for the small network (inner layer size $N = 100$), and 50–100 zJ (0.5–1 photon) for the large network (inner layer size $N = 1000$). Without being bound by any particular theory, this bound stems from the standard quantum limit (SQL): the intrinsic uncertainty of quadrature measurements on coherent states, which is temperature- and device-independent. This should be viewed as an absolute lower bound for the energy consumption of neural networks of this type; although the use of squeezed light allows one to reach sensitivity below the SQL, this involves squeezing all inputs (including vacuum inputs) which may lead to a net increase in overall energy consumption.

[0076] The SQL is network-dependent, and not all layers contribute equally. For each MAC, we have $\text{SNR} \propto n_{\text{mac}}$; however, the signal adds linearly while the errors add in quadrature. As a result, the larger network is more resilient to individual errors because each output is averaging over more neurons. Moreover, the solid curves in FIG. 3C are restricted to the case when n_{mac} is the same for all layers. The dashed lines show the error rate in a fictitious device where quantum-limited noise is only present in a particular layer. For the large network, a smaller n_{mac} can be tolerated in the second layer, suggesting that slightly better performance could be achieved by tuning the energy for each layer.

Energy Budget

[0077] FIGS. 4A–4C illustrate how viewing the neural network as an analog system with quantum-limited performance shifts the paradigm for comparing neural networks. FIG. 4A shows the standard approach: a scatterplot comparing error rate with number of MACs, a rough proxy for time or energy consumption, for different fully-connected MNIST neural networks. There is a tradeoff between size and accuracy, with larger networks requiring more operations but also giving better accuracy.

[0078] FIG. 4B plots the error rate as a function of photon number for the same neural networks as in FIG. 4A. Each point in FIG. 4A becomes a curve in FIG. 4B because it is possible to vary the number of photons per MAC, and the energy bound is set by the total number of photons, not

the number of MACs. While the general tradeoff between energy and accuracy is preserved, there are a number of counterintuitive results. For example, according to FIG. 4A, networks 1 and 2 have similar performance but the network 1 uses eight times more MACs, so under a conventional analysis, network 2 would always be preferred. However, FIG. 4B indicates that network 1 has better performance at all energy levels. This is because network 1 is less sensitive to shot noise due to averaging over many neurons, and therefore can be operated at lower energies, compensating for the increased neuron count. The same apparent paradox is seen with networks 3 and 4. This suggests that, in a quantum-limited scenario, reducing total energy may not be as simple as reducing the number of operations. Besides, the total energy may will depend on many other factors besides the SQL.

[0079] FIG. 4C is a plot of energy per MAC as a function of the average number of input neurons per layer N , a rough “size” of the neural network. The SQL data are plotted for the eight networks in FIGS. 4A and 4B, and the corresponding dashed line is an empirical fit. SQL dots correspond to minimum E_{mac} to make the error rate $p_{err}(E_{mac}) < 1.5p_{err}(\infty)$ (error bars correspond to $p_{err}(E_{mac}) = [1.2, 2.0]p_{err}(\infty)$). $E_{mac} = n_{mac}(hc/\lambda)$, $\lambda = 1.55\mu\text{m}$. The SQL is an absolute lower bound, assumes perfect detectors, and only counts input optical energy. In a realistic device, this curve may be shifted up by a factor $(\eta_d\eta_c\eta_s\beta_{mod})^{-1}$, where η_d , η_c and η_s are the detector, coupling, and source (laser) efficiencies and β_{mod} is the modulator launch efficiency.

[0080] The detector electronics also affect the energy budget. The homodyne signal from each neuron is sent through a nonlinear function $y_i \rightarrow f(y_i)$ and converted to the optical domain using a modulator or by modulating a light source directly (e.g., as in FIG. 2B). One way to do this is to digitize the signal, perform the function $f(\cdot)$ in digital logic, serialize the outputs, convert back to analog, and send the analog signal into the modulator. ADCs in the few-pJ/sample regime are available and simple arithmetic can be performed at the pJ scale. A range of modulator designs support few-fJ/bit operation. Thus, a reasonable near-term estimate would be few-pJ/neuron; this figure is divided by the number of inputs per neuron to give the energy per MAC (top sloped curve in FIG. 4C). A much more aggressive figure of 1 fJ/neuron is also plotted in FIG. 4C—approaching this bound may be possible with state-of-the-art modulators, but this bound is below the energy figures for ADCs. It may be achievable using well-designed analog electronics. At these energies,

the SQL becomes relevant, but as mentioned above, due to optical inefficiencies the SQL may be relevant at much higher energies as well.

[0081] For context, the ~ 1 pJ/MAC figure from state-of-the-art ASICs is shown in FIG. 4C (top trace). Energy consumption in non-reversible logic gates is bounded from below by the Landauer (thermodynamic) limit $E_{\text{op}} = kT \log(2) \approx 3 \text{ zJ}$. While multiply-and-accumulate is technically a reversible operation, realistic computers perform this operation using non-reversible binary gates, so Landauer's principle applies. A 32-bit multiplication involves approximately 10^3 binary gates and each bit operation consumes at least $kT \log(2)$, giving a limit $E_{\text{mac}} \geq 3 \text{ aJ}$ (dotted line in FIG. 4C). This is already higher than the SQL for the larger networks with $N \geq 100$. The optical neural network can achieve sub-Landauer performance because the matrix product is performed through optical interference, which is reversible and not subject to the Landauer bound.

[0082] A final consideration is the energy required to generate the weights in the first place. There is one weight pulse per MAC, so at the minimum this may be 1 fJ/MAC for the modulator and may rise above 1 pJ/MAC once the driver electronics and memory access are included. However, once the optical signal is generated, it can be fanned out to many neural networks in parallel, reducing this cost by a factor of B , the batch size. Large batch sizes should enable this contribution to E_{mac} to reach the few-femtojoule regime, and potentially much lower.

Training and Convolutions with Optical GEMM

[0083] As discussed above, the optical unit in FIG. 2B computes a matrix-vector product. Running multiple units in parallel with the same set of weights yields a general matrix-matrix product (GEMM), a function in the Basic Linear Algebra Subprograms (BLAS).

[0084] FIG. 5A shows a schematic for an optical GEMM unit 500 based on homodyne detection inspired by the neural-network concept illustrated in FIG. 2B. The inputs are two matrices $(M_1)_{m \times k}$ and $(M_2)_{n \times k}$, encoded into optical signals on the 1D transmitter arrays 510a and 510b, and mapped with cylindrical lenses (not shown) and a bulk optical beam splitter 520 to rows (columns) of a 2D detector array 530.

[0085] FIG. 5B shows a 1D transmitter array 510 in greater detail. The 1D transmitter array includes a waveguide 512 that receives an optical pulse train from a pulsed laser, modulated continuous-wave laser, or other suitable source. The waveguide branches, splitting the optical

pulse train into multiple copies. At the same time, routing and digital-to-analog conversion (DAC) circuitry 514 converts digital input signals into analog signals for driving modulators 516 at the outputs of the branched waveguide 512. The 1D transmitter array 510 emits the resulting modulated optical pulses as matrix elements M_{ij} that are arrayed in one spatial dimension, indexed by i , and in time, indexed by j .

[0086] From the accumulated charge at each pixel in the detector array 530, one can extract the matrix elements of the product $(M_1 M_2^T)_{m \times n}$. This operation uses $m \cdot n \cdot k$ MACs, with a total energy consumption (and energy per MAC) of:

$$E_{\text{tot}} = (mk + nk)E_{\text{in}} + (mn)E_{\text{out}}, \quad E_{\text{mac}} = \left(\frac{1}{n} + \frac{1}{m}\right)E_{\text{in}} + \frac{1}{k}E_{\text{out}} \quad (4)$$

where E_{in} , E_{out} are the transmitter and receiver energy use, per symbol, which include all optical energy plus electronic driving, serialization, DAC/ADC, etc. If the matrix dimensions (m, n, k) are large, significant energy savings per MAC are possible if $E_{\text{in}}, E_{\text{out}}$ can be kept reasonably small.

[0087] FIG. 5C shows matrix operations for inference, training, and back-propagation in a deep neural network that can be executed by the optical system 500 in FIG. 5A. When running a batch of B instances $X = [x_1 \dots x_B]$, the output $Y = [y_1 \dots y_B]$ can be computed through the matrix-matrix product $Y = AX$. In fully-connected layers, training and back-propagation also rely heavily on GEMM. The goal of training is to find the set of weights $A^{(k)}$ that minimize the loss function L , which characterizes the inaccuracy of the model. Training typically proceeds by gradient-based methods. Since the loss depends on the network output, training starts at the final layer and works backward in a process called back-propagation. At each layer, the optical processor 500 computes the gradient $(\nabla_A L)_{ij} = \partial L / \partial A_{ij}$ from the quantity $(\nabla_Y L)_{ij} = \partial L / \partial Y_{ij}$, and propagates the derivative back to the input $(\nabla_X L)_{ij} = \partial L / \partial X_{ij}$ (Fig. 5(c)). These derivatives are computed from the chain rule and can be written as matrix-matrix multiplications:

$$\nabla_A L = (\nabla_Y L) X^T, \quad \nabla_X L = A^T (\nabla_Y L) \quad (5)$$

Once the derivative has been propagated to $\nabla_{X^{(k)}} L$ (for layer k) we use the chain rule to compute $\nabla_{Y^{(k-1)}} L = f'(\nabla_{X^{(k)}} L)$ and proceed to the previous layer. In this way, the optical processor 500 sequentially computes the derivatives $\nabla_{A^{(k)}} L$ at each layer in the neural network.

[0088] In addition to fully-connected layers, it is also possible to run convolutional layers on the optical GEMM unit by employing a technique called patching. In a convolutional layer, the input $x_{ij;k}$ is a $W \times H$ image with C channels. This is convolved to produce an output $y_{ij;k}$ of dimension $W' \times H'$ with C' channels :

$$y_{ij;k} = \sum_{i',j',l} K_{i'j',kl} x_{(s_x i + i')(s_y j + j');l} \quad (6)$$

Here $K_{i'j',kl}$ is the convolution kernel, a 4-dimensional tensor of size $K_x \times K_y \times C' \times C$, and (s_x, s_y) are the strides of the convolution. Naïvely vectorizing Eq. (6) and running it as a fully-connected matrix-vector multiply is very inefficient because the resulting matrix is sparse and contains many redundant entries.

[0089] FIG. 5D illustrates a patching process to recast a convolution ($K_x = K_y = 3, s_x = s_y = 2$ shown) as a matrix-matrix multiplication. Patching expresses the image as a matrix X of size $K_x K_y C \times W' H'$, where each column corresponds to a vectorized $K_x \times K_y$ patch of the image. The elements of the kernel are rearranged to form a (dense) matrix K of size $C' \times K_x K_y C$. Eq. (6) can then be computed by taking the matrix-matrix product $Y = KX$, which has size $C' \times W' H'$.

[0090] On virtually any microprocessor, GEMM is a highly-optimized function with very regular patterns of memory access; the benefits of rewriting the convolution as a GEMM greatly outweigh the redundancy of data storage arising from overlapping patches. The time to rearrange the image as a patch matrix is typically very small compared to the time to compute the GEMM; therefore, by accelerating the GEMM, the optical matrix multiplier significantly increases the speed and energy efficiency of convolutional layers.

[0091] Since the optical processor performs the convolution as a matrix-matrix (rather than matrix-vector) operation, it is possible to obtain energy savings even without running the neural network on large batches of data. Computing the convolution requires $W' H' K_x K_y C' C$ MACs. Following Eq. (4), the energy per MAC (not including memory rearrangement for patching) is:

$$E_{\text{mac}} = \left(\frac{1}{C'} + \frac{1}{W' H'} \right) E_{\text{in}} + \frac{1}{K_x K_y C} E_{\text{out}} \quad (7)$$

$\frac{1}{c_{\text{in}}}$ $\frac{1}{c_{\text{out}}}$

[0092] The coefficients $c_{\text{in}} = (1/C + 1/W' H')^{-1}$ and $c_{\text{out}} = K_x K_y C$ govern the energy efficiency when the optical processor 500 is limited by input/output energies (transmitter/receiver

and associated electronics). Since reading a 32-bit register takes $\sim$ pJ of energy, a reasonable lower bound for near-term systems is $E_{in}, E_{out} \gtrsim$ pJ. Thus, the optical processor 500 should have $c_{in}, c_{out} \gg 1$ for its energy performance to beat an ASIC ($\sim$ pJ/MAC).

Layer	Input	Output	Kernel	Stride	MACs	c_{in}	c_{out}
CONV1	$227 \times 227 \times 3$	$55 \times 55 \times 96$	$11 \times 11 \times 96 \times 3$	4	105M	93	363
(pool)	$55 \times 55 \times 96$	$27 \times 27 \times 96$	–	2	–	–	–
CONV2	$27 \times 27 \times 96$	$27 \times 27 \times 256$	$5 \times 5 \times 256 \times 96$	1	448M	189	2400
(pool)	$27 \times 27 \times 256$	$13 \times 13 \times 256$	–	2	–	–	–
CONV3	$13 \times 13 \times 256$	$13 \times 13 \times 384$	$3 \times 3 \times 384 \times 256$	1	150M	117	2304
CONV4	$13 \times 13 \times 384$	$13 \times 13 \times 384$	$3 \times 3 \times 384 \times 384$	1	224M	117	3456
CONV5	$13 \times 13 \times 384$	$13 \times 13 \times 256$	$3 \times 3 \times 256 \times 384$	1	150M	102	3456
(pool)	$13 \times 13 \times 256$	$6 \times 6 \times 256$	–	2	–	–	–
FC1	$6 \times 6 \times 256$	4096	–	–	38M	–	–
FC2	4096	4096	–	–	17M	–	–
FC2	1000	1000	–	–	4M	–	–
	1.08G	132	1656				
	59M	–	–				

TABLE 1: Layers in AlexNet. Values of c_{in}, c_{out} are calculated from Eq. (7). Max-pooling layers after CONV1, CONV2 and CONV5 are used to reduce the image size, but the relative computational cost for these layers is negligible.

[0093] As a benchmark problem, consider AlexNet, shown in FIG. 6A, the first convolutional neural network to perform competitively at the ImageNet Large-Scale Visual Recognition Challenge. AlexNet includes five convolutional (CONV) layers and three fully-connected (FC) layers. Consistent with deep neural networks generally, the majority of AlexNet's energy consumption comes from the CONV layers. TABLE 1 above gives the layer dimensions and the values of c_{in}, c_{out} for the CONV layers in AlexNet. The MAC-weighted averages for all layers are $\langle c_{in} \rangle > 100$ and $\langle c_{out} \rangle > 1000$. Thus, even under extremely conservative assumptions of $E_{in}, E_{out} \gtrsim 100$ pJ (comparable to DRAM read energies), it is still possible to achieve sub-pJ/MAC performance. More advanced technology, such as few-fJ optical interconnects, may significantly reduce E_{in} and E_{out} , and therefore the energy per MAC. However, the performance is still fundamentally limited by detector shot noise (e.g., Eq. (3) for FC layers).

[0094] Using a pre-trained AlexNet model, FIG. 6B shows the top-10 accuracy on the ImageNet validation set as a function of the number of photons per MAC n_{mac} . Consistent with FIG. 3C, there are two limits: $n_{\text{mac}} \ll 1$ corresponds to the random guess regime with 99% error rate (for top-10 accuracy with 1,000 classes), while $n_{\text{mac}} \gg 1$ recovers the accuracy of the noiseless model.

[0095] The dashed lines in FIG. 6B show the (fictitious) case where noise is present in only a single layer, while the solid line corresponds to the case where all layers have noise and n_{mac} is the same for each layer. Not all layers contribute equally to the noise: CONV1 is the most sensitive, requiring $n_{\text{mac}} \gtrsim 20$, while the deeper layers (particularly the fully-connected layers) can tolerate much lower energies $n_{\text{mac}} \gtrsim 1$. Since the SNR is related to the total power received, which scales as $c_{\text{out}} n_{\text{mac}}$ for the convolutional layers (c_{out} pulses per detector), it is not surprising that the deeper layers, which have a larger c_{out} , are less sensitive to quantum noise. The SQL obtained for AlexNet ($n_{\text{mac}} \gtrsim 20$ or $E_{\text{mac}} \gtrsim 3$ aJ) is slightly larger than that from the MNIST networks in FIG. 3C, but of the same order of magnitude, suggesting that the SQL is somewhat problem-dependent.

Discussion

[0096] This architecture for optically accelerated deep learning is scalable to large problems and can operate at high speeds with low energy consumption. It takes advantage of the photoelectric effect, via the relation $I \propto |E|^2$, to compute the required matrix products opto-electronically, obviating the need for all-optical nonlinearity that has hobbled past approaches to optical computing. Since the device can be constructed with free-space optical components, it can scale to much larger sizes than nanophotonic implementations, being ultimately limited by the size of the detector array (e.g., $N \gtrsim 10^6$).

[0097] One advantage to this optical processor is that the multiplication is performed passively by optical interference, so the main speed and energy costs are associated with routing data into and out of the optical processor. For a matrix multiplication $C_{m \times n} = A_{m \times k} B_{n \times k}$, the input/output (I/O) energy scales as $O(mk) + O(nk) + O(mn)$, while the number of MACs scales as $O(mnk)$. For moderately large problems found in convolutional neural-network layers (e.g., $m, n, k \geq 100$) with moderate input/output (I/O) energies (e.g., $\sim$ pJ), performance in the ~ 10 fJ/MAC range should be feasible, which is 2–3 orders of magnitude smaller than state-of-the-art CMOS circuits. Advances in optical interconnects may reduce the I/O energies by large factors, translating to further reductions in energy per MAC.

[0098] The fundamental limits to optical processors affect their long-term scaling. For the optical neural network presented here, detector shot noise presents a standard quantum limit (SQL) to neural network energy efficiency. Because this limit is physics-based, it cannot be engineered away unless non-classical states of light are employed. Monte Carlo simulations of pre-trained models for MNIST digit recognition (fully-connected) and ImageNet image classification (convolutional) show that optical neural network performance is a function of the number of photons used, which sets a lower bound on the energy per MAC. This bound is problem- and network-dependent and lies in the range 50 zJ–5 aJ/MAC. By contrast, the Landauer limit for electronic neural networks is 3 aJ/MAC (assuming 1,000 bit operations per MAC); sub-Landauer performance is possible in optical neural networks because the multiplication is performed through optical interference, which is reversible and not bounded by Landauer’s principle.

2. Digital Optical Neural Networks with Holographic Optical Fan-Out Interconnects

[0099] Convolutional neural networks (CNNs) are a key tool in machine learning. As neural networks grow larger and deeper, so do the energy requirements of the convolutional layers. Even small CNNs (by 2018 standards) like AlexNet use nearly 0.5 GMACs per classification step, and more modern CNNs use orders of magnitude more. Thus, there is strong motivation to find special-purpose hardware to both speed up and increase the energy efficiency of CNNs.

Taxonomy of Convolutional ONNs

[00100] The section above describes an analog optical neural network (ONN) based on homodyne imaging. This section describes a digital optical CNN. In a conventional processor, both the logic and the interconnections are done in electronics. The digital ONN retains the electronics for logic, but does interconnection using optics. Holography and Fourier optics are especially well-suited to realize the data and weight fan-out in a convolutional network, which reduces the number of memory-access calls per MAC by a large factor, significantly reducing the total energy consumption.

[00101] TABLE 2 lists differences between electronic CNNs, digital ONNs, analog ONNs, and coherent ONNs. An analog ONN uses analog circuits in place of the digital logic in a digital ONN. Because energy consumption in the digital ONN is limited by the arithmetic logic units (ALUs), replacing digital logic with analog circuits can deliver several orders of magnitude of speed and efficiency gains despite analog noise and calibration issues. While analog, this system

is still optically incoherent and relies on the optics only for communication. The most powerful neural network is the coherent ONN, which performs both logic and communication optically, relying on electronics merely as a buffer for integration and readout. The coherent system beats the energy consumption analog and digital ONN by several orders of magnitude; however, it is the most complex of the three systems and uses coherent sources and detection.

	GPU / TPU	ONN (digital)	ONN (analog)	ONN (coherent)
Logic	Digital ALUs	Digital ALUs	Analog electronics	Optical (homodyne)
Interconnects	Wires	Optical (incoherent)	Optical (incoherent)	Optical (coherent)
Energy/MAC	$\gtrsim 20$ pJ	< 1 pJ	$\gtrsim$ fJ	aJ–fJ

TABLE 2: Comparison between different CNN hardware platforms.

[00102] Before describing the digital ONN in greater detail below, it is useful to discuss the differences between fully-connected (FC) and convolutional (CONV) layers so that one can glean insight into how optical connections can save energy. An FC layer has all-to-all connectivity between neurons: $y_i = \sum_j A_{ij} x_j$. Therefore, one desires a broadcast connection (one-to-all) for the data, while the weight connections are one-to-one (Sec. 1). Significant savings are possible by routing the data optically, but most of the energy in FC layers comes from accessing the weights; reducing this contribution depends on careful engineering of the weight server and running multiple neural networks in parallel, amortizing the cost of the weights.

[00103] A convolutional layer, by contrast, implements a far more structured map:

$$y_{ij;m} = \sum_{kl;n} (K_{kl;mn} x_{i+k,j+l;n}) \quad (8)$$

$$(W' \times \tilde{H}' \times C') \quad (K_x \times K_y \times \tilde{C}' \times C) \quad (W \times \tilde{H} \times C)$$

[00104] Here (k, l) are the x- and y-indices of a pixel, and m is the channel index. Typical dimensions are $W, H, W', H' = 10\text{--}50$, $K_x, K_y = 3\text{--}5$, and $C, C' = 100\text{--}400$ in the hidden layers. Each input $x_{ij;m}$ is fanned out to $K_x K_y C'$ outputs, while each weight is fanned out $W' H'$ times. This is an extraordinary amount of redundancy, both for inputs and for weights. Therefore, significant performance improvements are possible if one can passively perform the required fan-

out. As I will show in the following section, free-space optics can be adapted to precisely this end, both in the digital and analog (incoherent) cases.

Interconnects for a Digital ONN

[00105] FIG. 7A shows a single CONV layer for a CNN. The convolution Eq. (8) is represented as a weighted fan-in to a single output pixel y_{ij} as shown in FIG. 7B. By flipping the kernel around, the convolution can equivalently be viewed as a fan-out from a single pixel x_{kl} as shown in FIG. 7C. The fan-out picture is helpful for designing an optical interconnect that reduces or minimizes the number of memory calls (either from inputs or weights) per calculation.

Patch Method for Input Fan-Out: Maximum Input Fan-Out

[00106] Each input maps to $K_x K_y C'$ outputs. This fan-out can be achieved if each input pixel can be mapped to a $K_x \times K_y$ patch of output pixels, with each output pixel comprising C' detectors (sub-pixels), one for each channel. This effectively down-samples the image, and channel data is sent in sequentially, so $K_x K_y C$ clock cycles are used to transfer the data, but since the total number of MACs is $K_x K_y W' H' C' C$ and approximately $(W'/K_x)(H'/K_y)C'$ MACs are performed per clock cycle, the ALUs are operating at near 100% utilization in this scheme, which is very efficient.

[00107] FIG. 8 illustrates this patch method 800 and its implementation with a Fourier optical system shown in FIGS. 9A and 9B and described below. In this method 800, a $(W \times H) = 9 \times 9$ image is convolved with a $(K_x \times K_y) = 3 \times 3$ kernel. First, the image is broken into patches of size $K_x \times K_y$ (802). Rather than firing all inputs simultaneously, only one pixel fires per patch in a given “frame.” To transmit the data, the CONV layer uses $K_x \times K_y$ frames, and each frame includes C steps, where C is the channel dimension.

[00108] Following Eq. (8) and FIG. 7C, for maximum fan-out, the active pixel (i, j) is mapped to the $K_x \times K_y$ patch that intersects (i, j) at the lower-right corner, that is, $(i - k, j - l)$ for all $k < K_x, l < K_y$. To do so, the transmitter output is convolved with a box function that spreads the field evenly over all pixels in the patch (804). The convolution causes all pixels to fill up their respective patches, resulting in a “down-sampled” image 806.

[00109] Each $K_x \times K_y$ patch is weighted with a (flipped) kernel (e.g., as in FIG. 7C). Therefore, the kernel is accordingly convolved with a lattice of delta-functions with x- and y-spacing K_x and K_y , respectively (814). This convolution causes the $K_x \times K_y$ kernel to tile the

image (816). Care should be taken to ensure that the kernel tiles line up with the input patches; thus, each frame should input an appropriately shifted kernel as shown in FIG. 8.

[00110] The input and kernel signals are imaged onto a detector chip (808), which has $W' \times H'$ pixels with C' sub-pixels per pixel and within each sub-pixel, an ALU to do MACs, and one detector each for $x_{ij;n}$ and $K_{kl;mn}$. For each step, within a given pixel the x sub-pixel inputs are the same (broadcast of optical data), while the K sub-pixel inputs are different. During frame (k, l) , each pixel (i, j) is performing an all-to-all matrix-vector multiply between matrix $K_{kl;mn}$ and vector $x_{i+k,j+l;n}$ (Eq. (8)). Within each pixel, the process is identical to the digital ONN for FC layers disclosed above.

	D ² NN	Fourier ONN (Sec. 3)	Digital ONN
Encoding	Analog	Analog	Digital
Coherence?	Yes	Yes	No
Inputs $x_{ij;m}$	Optical	Optical	Optical
Kernel $K_{ij;mn}$	SLMs / diffraction layers	Optical	Optical
Use of Fourier optics	Perform convolution	Perform convolution	Fan out data
# channels	1	$\gg 1$	$\gg 1$

TABLE 3: Comparison between D²NN, a coherent Fourier ONN, and a digital ONN

[00111] A convolution can be implemented with two Fourier transforms and a filter:

$$A^* \otimes B = \sqrt{2\pi} \mathcal{F}^{-1}[\mathcal{F}(A)^* \mathcal{F}(B)] \quad (9)$$

[00112] Since the convolutions in the method 800 of FIG. 8 are taken on spatially-encoded optical data, they can be performed using Fourier optics: convert the image to its Fourier transform with a lens, pass the Fourier-transformed image through an SLM, and perform an inverse Fourier transform on the SLM output with another lens. This looks deceptively similar to the recent “diffractive deep neural network” (D²NN) because both networks perform convolutions and use Fourier optics. However, the similarities end there. In the D²NN, the kernel is hard-coded into the diffraction layers; by contrast, the digital ONN disclosed here encodes the kernel optically, which allows it to be changed on the fly and shared among multiple “client” networks if a batch of data is being processed in with the same network. Another distinction is that although the D²NN and the Fourier ONN both use the optical Fourier transform to perform the convolution (albeit in

different ways), a digital ONN uses the optical Fourier transform only to fan out the data. Because the data in for the digital ONN is encoded digitally, Fourier optics, which are designed for coherent analog signals, would not perform the convolution correctly.

[00113] Performing the optical fan-out efficiently can be tricky in practice. If the phase of the light should be constant, there will be an N -fold power reduction when fanning out one mode to N modes. This can be seen by inputting a plane wave (all pixels on with same amplitude). Since the convolution of a constant is also a constant, without the factor-of- N hit, the output power would be greater than the input power. In terms of Eq. (9), the input light is a series of tightly confined dots with most power at large spatial frequencies. This power is filtered out by the box convolution, which is a tightly-peaked sinh. Fortunately, the patch method 800 for the digital ONN does not rely on the optical phase. Relaxing the phase constraints, some results from Fourier holography can be used here.

[00114] FIG. 9A illustrates the basic principle of Fourier holography in a $4f$ optical system, where f is the focal length of the lenses and $4f$ refers to the total system length. A point source 902 is mapped to a plane wave using a confocal lens 904, and the amplitude and/or phase of that wave is modified using a spatial filter or SLM 906 in the Fourier plane 907 of the confocal lens 904. A second lens 908 Fourier-transforms this waveform into an image (the Fourier hologram 910). It is possible to produce a Fourier hologram with high image quality and good efficiency by using a quadratic (Guoy) or random phase pattern with error diffusion. FIG. 9B shows that the effect is the same as a convolution for multiple point sources 912 as long as the resulting holograms 920 do not overlap. Thus, the fan-out convolutions in FIG. 8 can be performed all-optically. The fan-out kernels are static, can be pre-computed, and depend only on the size of the kernel $K_{ij,mn}$, not its values. Therefore, the SLM state can be changed infrequently (e.g., when the kernel size changes), so is unlikely to affect the computation time or energy consumption.

[00115] While the input fan-out is extremely efficient, the kernel fan-out is less so. Each kernel weight is called $W'H'$ times, however, the fan-out in FIG. 8 is $(W'/K_x)(H'/K_y)$. While still ≥ 20 for typical CNN layers, this is much less than the input fan-out ratio. To reduce it further, the CONV layer can be run on multiple datasets at once, increasing the potential fan-out by the batch size.

[00116] Another complication arises from the kernel “shifting” done to ensure the kernel patches lined up with the data patches. If done electronically, this may incur a large communication cost if the chip is not sufficiently small (the amount of memory required for weights is actually quite large – K_x and K_y are minimal, but C and C' can be in the hundreds). The shifting could be done optically using programmable nanophotonic processors (PNPs) or a fast beam-steering SLM if one can be developed, but these technologies are in their infancy and may be power-hungry (and there are many modes to shift: $K_x K_y C'$ in all). The added energy costs of shifting should be sufficiently diluted by the fan-out that they become irrelevant.

Shift Method for Input and Kernel Fan-Out: Maximum Kernel Fan-Out

[00117] FIG. 10 illustrates a shift method 1000 for fanning out inputs and kernels that maximizes kernel fan-out. As before, the calculation is broken down into $K_x K_y$ frames, with each frame including C steps, one for each input channel (1002). Each frame corresponds to a pair (k, l) in Eq. (8). Each frame is convolved with a box function (1004), resulting in a “down-sampled” image 1006, which is weighted with a kernel.

[00118] Each kernel element $K_{kl,mn}$, which is a $C' \times C$ fully-connected matrix, is convolved with a lattice of delta-functions (1014) and broadcast to all pixels during the frame (there are C steps per frame and C' sub-pixels per pixel, allowing $C'C$ matrix elements to be transferred during the frame) for detection by a detector array (1008). Thus, the weight fan-out is $W'H'$, the maximum possible fan-out (1016). Typically, this is around 100–2000 for CONV layers. As in the patch method 800 (FIG. 8), this fan-out can be achieved all-optically using Fourier holography, e.g., using the Fourier optical system shown in FIGS. 9A and 9B with the SLM 906 set to produce the desired shifts for the input and kernel fan-out.

[00119] The price for maximizing kernel fan-out is a reduced fan-out for the inputs and some added complexity in sending them in. The input fan-out is C' rather than $K_x K_y C'$ since the pixels are mapped one-to-one (convolution and fan-out still happen because each input maps to all C' sub-pixels). However, the channel number is large in hidden CONV layers, so this is typically hundreds.

[00120] In addition, the shift method 1000 uses shifted images rather than the originals. The shifting can be done electronically at a cost of higher energy per bit (because we need to send data

between pixels in each step). However, reading data from a Network-on-Chip (NoC) or local buffer is usually only 2–6 times more costly than reading from a register, so fan-out should win out over this added cost.

[00121] Finding an efficient optical steering mechanism to do the shifts is not essential to this scheme (or the patch method 800 in FIG. 8). However, it would allow some benefits: local memory accesses for the $x_{ij;m}$, and if the sum in Eq. ([eq:s3-conv]) is done in the right order (iterate over (k, l) in the inner loop, n in the outer loop), the effective fan-out increases to $K_x K_y C'$ because the modulator is only set once at the beginning of the $K_x K_y$ consecutive pulses (for channel n). (Likewise, one can increase the kernel fan-out in the patch method 800.) Several ways to optically steer an image are programmable PNPs or beam-steering SLMs. One- or two-dimensional steering can also be achieved with a frequency-swept source and one- or two-dimensional grating or virtually image phased array.

[00122] The initial layers of CNNs down-sample using a stride $s_x, s_y > 1$. Thus, the convolution function becomes:

$$y_{ij;m} = \sum_{kl;n} K_{kl;mn} x_{s_x i + k, s_y j + l; n}$$

[00123] Stride reduces the output size (W', H') as well as fan-out because we are no longer taking a convolution over every $K_x \times K_y$ block. While the kernel fan-out is unaffected, input fan-out is reduced to $\lceil K_x/s_x \rceil \lceil K_y/s_y \rceil C'$ for the patch method 800 of FIG. 8. The fan-out in the shift method 1000 is unaffected because each input is fanned out to one pixel in each step anyway. Stride also scrambles the order of the shifted kernels in the patch method 800 of FIG. 8, but this can be pre-computed.

	Input fan-out	Kernel fan-out	Steps	Complexity
Theoretical Limit	$\lceil K_x/s_x \rceil \lceil K_y/s_y \rceil C'$	$W' H'$	–	–
Patch Method	$\lceil K_x/s_x \rceil \lceil K_y/s_y \rceil C'$	$W' H' / K_x K_y$	$K_x K_y C$	Kernel shifting
Shift Method	C'	$W' H'$	$K_x K_y C$	Input shifting

TABLE 4: Comparison of patch and shift methods for optical fan-out.

Incoherent Analog ONN

[00124] The same optical fan-out can be ported to an incoherent analog ONN, which may be orders of magnitude more efficient than a digital ONN. In the best-case scenario, the memory-access costs can be rendered negligible. Since this comprises 80–90% of the energy in CNNs like AlexNet, as shown in FIG. 11A, the energy reduction may be only a ten-fold. A floating-point operation at 32-bit involves around 1,000 gate operations and consumes about 1 pJ. By extrapolation, if the operation were performed in analog, with one or two transistors, it should consume $O(fJ)$.

[00125] FIG. 11B shows a simple analog circuit that implemented a weighted MAC: $\sum_i f(K_i)x_i$. The top photodetector charges the transistor that gates the signal from the bottom photodetector. The function $f(\cdot)$ is highly nonlinear and depends on the leakage resistances, but by inverting it and sending an input $f^{-1}(K_i)$ one can obtain the desired product. A simple extension of this circuit allows both K and x to be negative, at the cost of doubling the number of circuit elements.

[00126] CMOS photodetectors and reasonably short wires (e.g., about 10 μm in length) have a capacitance of several fF, so fJ/MAC seems to be a lower bound in the incoherent analog case. Whether this bound is reached in practice depends on the memory-access costs. The overall energy per MAC, E_{mac} , can be estimated with the following equation:

$$E_{mac} = E_{logic} + E_{pd} + \left(\frac{1}{F_x} + \frac{1}{F_K}\right)(E_{mem} + E_{mod} + E_{dac}) \quad (11)$$

[00127] The various energy costs are tabulated below in TABLE 5.

Name	Description	Value (digital) [‡]	Value (analog)	Notes
E_{logic}	Logic gate	$(n/32)^2 \times pJ$	fJ	See Refs.
E_{pd}	Photodetection*	$(n/32) \times 32 \text{ fJ}$	fJ	fJ/bit
E_{mem}	Memory read [†] (register)	$(n/32) \times pJ$	(??)	30 fJ/bit
E_{mod}	Modulator	$(n/32) \times 32 \text{ fJ}$	fJ	fJ/bit
E_{dac}	DAC	0	pJ (??)	
F_x, F_K	Fan-out for $x_{ij;m}, K_{ij;mn}$			

TABLE 5: Parameters in Eq. (11). *Optical plus electrical (receiver) power. [†]Read from register file. Non-register local memory is 2–6 times larger. [‡] n = number of bits.

[00128] Several factors make the analog approach more complicated. Process variations may make some transistors more responsive than other, leading to a non-uniform weighting function $f(K)$; this will need to be corrected for, perhaps by tuning the resistor values. Also, while the box/lattice convolution kernels which perform the optical fan-out in FIGS. 8 and 10 only needed to be specified approximately in a digital ONN, the signal should fan-out in a uniform manner for an analog ONN. Otherwise, the convolution function may become position-dependent, which could degrade performance (unless corrected). Put another way, the Fourier hologram image-quality requirements are higher in the analog case. This may reduce the hologram efficiency and may involve both amplitude and phase modulation, increasing system complexity.

3. Convolutional Neural Network Based on Fourier Optics

[00129] The ONN in Sec. 1 uses coherent (homodyne) detection to combine the weights and neuron activations, thus performing a matrix-vector multiplication to implement a fully-connected (FC) layer. However, convolutional (CONV) layers are used in many neural networks, especially in image processing. This section describes a modified ONN that uses combines Fourier optics and coherent detection to efficiently implement a CONV layer.

Convolutional Neural Networks

[00130] Convolutional neural networks are used to process data with structure, particularly images. For large feature vectors, convolutions are much faster than all-to-all matrix multiplications and have fewer degrees of freedom, which speeds up training.

[00131] In a convolutional network, the data is represented as a 3-dimensional ($W \times H \times C$) array: $x_{ij;m}$. Here (i, j) are the physical coordinates, while m is the channel index. Equivalently, one can view the data as a collection of C feature maps, each of dimension $W \times H$. Each CONV layer convolves the feature-map array with a kernel $K_{kl;mn}$ to produce the synaptic inputs:

$$y_{ij;m} = \sum_{kl,n} K_{kl;mn} x_{i+k,j+l;n} \quad (12)$$

[00132] The kernel has a dimension ($K_x \times K_y \times C' \times C$), giving the output $y_{ij;m}$ the shape ($W' \times H' \times C'$), where $W' = W - K_x + 1$, $H' = H - K_y + 1$. A nonlinear function (e.g., sigmoid, ReLU, etc.) maps each y to the neuron activations: $x_{ij;m} = f(y_{ij;m})$. These are the feature-maps input to the next layer of the network.

[00133] For a single input and output channel, Eq. (12) is a simple convolution. However, the feature-maps in real neural networks have many channels, e.g., $C, C' \gg 1$. K acts as a matrix multiplication in the channel index m , and as a 2D convolution for the coordinate indexes (i, j) .

[00134] In the deeper layers, the images are quite small, but involve a large number of channels (e.g., as in the AlexNet shown in FIG. 6B). The kernel matrix can be quite large, even though its dimension in (i, j) -space is only 3×3 . Nevertheless, the matrices are extremely sparse ($\sim 1\%$ of elements are nonzero) and very repetitive (each weight used ~ 280 times), so naively implementing the convolution as a dense matrix-vector multiplication is extremely inefficient (e.g., 118 GMAC vs. 1.08 GMAC) and the matrices (~ 0.5 TB total) may not fit into memory.

Layer	Feature-map	(size)	Kernel	(size)	MACs
CONV1	$227 \times 227 \times 3$	150k	$11 \times 11 \times 96 \times 3$	35k	105M
CONV2	$55 \times 55 \times 96$	290k	$5 \times 5 \times 256 \times 96$	615k	448M
CONV3	$27 \times 27 \times 256$	186k	$3 \times 3 \times 384 \times 256$	885k	150M
CONV4	$13 \times 13 \times 384$	65k	$3 \times 3 \times 384 \times 384$	1.3M	224M
CONV5	$13 \times 13 \times 384$	65k	$3 \times 3 \times 256 \times 384$	885k	150M
FC1	$6 \times 6 \times 256$	9k	4096×9216 (FC)	37M	37M
FC2	4096	4k	4096×4096 (FC)	16M	16M
FC3	4096	4k	1000×4096 (FC)	4M	4M

TABLE 6: Layers in AlexNet. Max pooling is done before FC1 to reduce $13 \times 13 \times 256$ feature-vector to $6 \times 6 \times 256$. Most of the weights are in the FC layers, while most of the MACs are in the CONV layers.

[00135] Several approaches can be taken to compute a convolution efficiently. One approach is to vectorize the kernel and convert the feature-map to a Toeplitz matrix (with redundant data). Another vectorization approach called patching has been implemented on GPUs and is amenable to all-optical computation with time-encoded data if delay lines are used, however, the total delay length needed is quite large and may not be practical in the near term.

Convolution via Fourier Optics

[00136] FIG. 12A illustrates a method 1200 for performing a convolution efficiently by Fourier transform. This is a common technique in signal processing and is particularly efficient for large kernels. It works in both discrete and continuous space. Define the convolution as:

$$\begin{aligned} (A \otimes B)_i &= \sum_l A_l B_{i+l} & (\text{discrete}) \\ [A \otimes B](x) &= \int A(y)B(x+y)dy & (\text{continuous}) \end{aligned} \quad (13)$$

[00137] With the symmetric normalization for the Fourier transform ($\mathcal{F}^{-1}$ obtained by flipping the sign in the exponent) (1202)

$$\begin{aligned} \mathcal{F}[A]_k &= \frac{1}{\sqrt{N}} \sum_l e^{2\pi i k l / N} A_l & (\text{discrete}) \\ \mathcal{F}[A](k) &= \frac{1}{\sqrt{2\pi}} \int e^{ikx} A(x) dx & (\text{continuous}) \end{aligned} \quad (14),$$

the convolution may be computed by an elementwise multiplication in Fourier space (1204) followed for an inverse Fourier transform (1206) back to real space:

$$\begin{aligned} A^* \otimes B &= \sqrt{N} \mathcal{F}^{-1}[\mathcal{F}(A)^* \mathcal{F}(B)] & (\text{discrete}) \\ A^* \otimes B &= \sqrt{2\pi} \mathcal{F}^{-1}[\mathcal{F}(A)^* \mathcal{F}(B)] & (\text{continuous}) \end{aligned}$$

[00138] In the digital electronic domain, this process 1200 uses three fast Fourier transforms (FFTs). Since an FFT involves $O(N \log N)$ steps while a straightforward approach (banded matrix-vector multiply) involves $O(NK)$ steps, the FFT is advantageous when the kernel is sufficiently large.

[00139] Optical diffraction in the Fraunhofer limit also implements a 2D Fourier transform. A convolution can be implemented by hard-coding the kernel with an SLM and using a pair of lenses to perform the Fourier transform, e.g., as shown in FIG. 9A. A 3D-printed “diffractive deep neural network” (D^2NN) based on this principle achieves reasonable performance for digit classification. However, filters of this sort only produce a single-channel convolution; scaling it to a large number of channels may be problematic. Moreover, the D^2NN is hard-wired to solve one and only one problem. A modified D^2NN using SLM’s would be programmable, but the programming speed would be slow and anyways it is not clear that the architecture will work for deep CNN’s which almost always have many channels ($C \gg 1$).

[00140] FIGS. 12B and 12C show a convolution (CONV) layer 1220 that implements a convolution using three Fourier transforms and a product with optically encoded weights and

coherent detection. FIG. 12B shows how weights and feature maps are Fourier-transformed and the product $\tilde{K}_{kl}^* \tilde{x}_{kl}$ is detected in the Fourier plane. FIG. 12C shows how that product $(\tilde{K}_{kl}^* \tilde{x}_{kl})^*$ is Fourier-transformed and coherently detected in the image plane, producing the convolution $(\mathcal{F}^{-1}[\tilde{K}^* \tilde{x}])^*$. Like the D²NN, this system uses the optical Fourier transform to perform the convolution. However, the weights are encoded optically and are thus tunable.

[00141] The CONV layer 1220 includes a coherent transceiver array 1230 in the image plane and a coherent transceiver array 1240 in the Fourier plane. The neurons reside on the image plane. A separate transmitter array 1250 (Weight Server) provides the kernel weights. A beam splitter 1270 and other optional optics (not shown) are arranged so that signals at the image-plane transceiver array 1230 and weight server 1250 are Fourier-transformed when they arrive at the Fourier-plane transceiver array 1240.

[00142] In the first step of the convolution, shown in FIG. 12B, the image-plane transceiver array 1230 and weight server 1250 fire. All transmitters fire simultaneously, producing the 2D Fourier transforms of the image and kernel that are detected by the Fourier-plane transceiver array 1240. For simplicity, suppose that the transceivers 1240 at the Fourier plane can distinguish between the weights and feature-map signals (they could come in at opposite angles to oppositely-oriented grating couplers, or pinhole arrays and appropriate imaging could be used to ensure the appropriate signals map to the appropriate gratings). Each pixel (transceiver 1240) in the Fourier plane measures the homodyne product $\tilde{K}_{kl}^* \tilde{x}_{kl}$. In this case, both real and imaginary components are stored (say by having two homodyne detectors per transceiver, 90 degrees out of phase, as shown in FIG. 13 and described below).

[00143] In the second step of the convolution, shown in FIG. 12C, the transceivers 1240 in the Fourier plane transmit the product $(\tilde{K}_{kl}^* \tilde{x}_{kl})^*$ back to the image plane. This product is produced using modulators and the real and imaginary parts stored in the first step of the convolution (FIG. 12B). Since the output at the Fourier plane is the conjugate of the desired field, the CONV layer 1220 does an inverse Fourier transform going back to the image plane. The homodyne signal relative to a local oscillator 1260 produces the desired convolution $K^* \otimes x$. Note that x and K are real and therefore the conjugate is irrelevant.

[00144] FIG. 13 shows a transceiver 1300 suitable for use in the Fourier-plane transceiver array 1240 of FIGS. 12B and 12C. In operation, the transceiver 1300 detects in-phase and

quadrature components of input signals (e.g., elements of the Fourier-transformed input and kernel) and emits an output equal to the product of the input signals (e.g., the product of the Fourier-transformed input and kernel elements). Gratings 1302 and 1312 couple light to a first and second 1×2 beam splitters 1304 and 1314, respectively, via waveguides integrated into a photonic integrated circuit. Each beam splitter 1304, 1314 sends one output to a quadrature beam splitter 1308 and one output to an in-phase beam splitter 1318. A phase shifter 1306 shifts the phase of one input to the quadrature beam splitter 1308 by $\pi/2$. The quadrature and in-phase beam splitters 1308, 1318 combine these outputs for detection by respective differential detectors 1310, 1320. The outputs of the differential detectors 1310, 1320 drive respective modulators 1332, 1336, which modulate copies of a local oscillator (LO) beam from the LO 1270 in FIGS. 12B and 12C. Another phase shifter 1334 shifts the phase of the output of the quadrature modulator 1336 by $-\pi/2$. The modulated LO beams are recombined and emitted through grating 1312 via a crossbar optical switch 1316.

Multiple Channels

[00145] The hidden layers of a convolutional neural network have relatively small image dimensions, but a very large number of channels, with the number of channels increasing deeper in the network. This feature of convolutional networks allows high-resolution spatial information to be collapsed into low-resolution “contextual” information enabling the detection of edges, polygons, and more complex features, such as faces.

Time Encoding

[00146] FIGS. 14A and 14B show time multiplexing schemes that can be used to encode channel information with the optics of FIGS. 12B, 12C, and 13. For a layer with C and C' input and output channels, respectively, there are a total of CC' Fourier transforms (on both signal and weights), in addition to CC' MACs at each Fourier-plane pixel. In this way, the Fourier-plane circuit obtains C' outputs $\sum_n \tilde{K}_{mn}^* \tilde{x}_n$. These outputs are then sent back to the image plane, performing inverse Fourier transforms one channel at a time, to obtain the multi-channel convolution in Eq. (12). Because this involves a total of CC' steps, each layer is a factor of CC' slower than the single-channel case, which is a very large factor (10^4 – 10^5) for the inner layers in convolutional networks.

[00147] This approach is also problematic because of the multiple read/write steps for each memory element. The problem exists regardless of whether the matrix is C-ordered, as in FIG. 14A, or Fortran-ordered, as in FIG. 14B. For C-ordering, each input x is read C' times, with one write step is needed per output (each pulse train is integrated until the end of the sequence and then read out). Conversely, Fortran ordering simplifies the input step (each modulator is set once for each x_m), but the readout to each memory is staggered and involves C writes per element. Accessing analog memory can be noisy because there is no error correction, and many reads/writes may lead to large errors. Conversely, accessing digital memory incurs DAC/ADC costs, which are $O(p)$ per sample, which is may be prohibitive.

Frequency Encoding

[00148] FIGS. 15A and 15B illustrate encoding the channels in frequency rather than time. This scheme takes advantage of wavelength-division multiplexing (WDM) technology to avoid multiple read/write operations as in the time-encoded scheme. As shown in FIG. 15A, an image-plane client 1502 sends out C' identical pulses on each WDM channel, where the amplitudes of the pulses for each channel are the inputs x_m . A single memory read sets the modulator. Similarly, a weight generator 1504 encodes each matrix column $A_{:,n}$ in the corresponding (n^{th}) channel. As a result, the output at time-step m is $\sum_n \tilde{K}_{mn} \tilde{x}_n$; the summation is done for free in a homodyne detector 1506 and the memory is written to once.

[00149] One downside with frequency encoding is the added complexity of the WDM channels. For the deeper layers of AlexNet with $C' = 384$, this is a very large number of channels, which may not be practical. However, existing WDM systems rely on fast modulators where the data rate per channel is $\gtrsim 25\text{Gbps}$, so the channel spacing is usually $\gtrsim 50\text{GHz}$. The C-band is 30 nm wide and accommodates 80 channels with 50 GHz spacing. By working at lower speeds (e.g., GHz), the C-band can accommodate more than 80 channels, e.g., using thermally stabilized high- Q filters.

[00150] FIG. 15B illustrates one way to realize a many-channel WDM on-chip. This uses a traditional WDM, such as an arrayed waveguide grating (AWG) or cascaded Mach-Zehnder filters 1552, to do the (de)multiplexing at coarse channels, while high- Q ring resonators 1554 provide the fine resolution. Each pixel (transceiver) has a separate WDM unit on each pixel, which may be practical for inner layers with hundreds of pixels.

[00151] Chromatic aberration presents a separate problem. The FFT performs the mapping

$$A_k \rightarrow \frac{1}{\sqrt{N}} \sum_l e^{2\pi i k l / N} \quad (17)$$

The maximum phase in the exponential is $O(N)$, occurring when $k = l = N - 1$. In optics, however, the phase is proportional to wavelength. If the wavelength is changed by $\Delta\lambda$, all phases in the optical FFT scale by $\phi \rightarrow (1 + \Delta k/k)\phi \approx (1 - \Delta\lambda/\lambda)\phi$. If the optical FFT has a phase tolerance of $\Delta\phi$, the tolerance on $\Delta\lambda$ will be:

$$\frac{\Delta\lambda}{\lambda} \lesssim \frac{\Delta\phi}{N} \quad (18)$$

Fortunately, N (the width/height of the image) is not that large. For an intermediate layer with $N = 27$ and $\Delta\phi = 0.1$, the wavelength tolerance is $\Delta\lambda \lesssim 6$ nm. This is about 1 THz, which is enough for several hundred channels, each spaced by a couple of GHz. Note the $\lesssim$ sign indicates there may be some $O(1)$ factor here too.

Spatial Encoding

[00152] It is also possible to encode the data in spatial modes. However, the principle here is more subtle and care should be taken to prevent the Fourier transform from distorting or degrading the spatial encoding. To start, consider several facts about Fourier transforms.

[00153] The Fourier transform of a comb is a comb (a sum of evenly spaced delta functions). Of relevance here are the Fourier transforms of functions that are nearly comb-like, but not quite. To begin, recall two definitions: (1) a function $f(x)$ is nearly periodic with period L if $|f(x + L) - f(x)| \ll f(x)$ and (2) function $f(x)$ is a spike train with period L if $f(x) \approx 0$ for all x unless $x \approx mL$ for $m \in \mathbb{Z}$.

[00154] The Fourier transform of a spike-train function with period L is a nearly-periodic function with period $2\pi/L$. To see why, consider a spike-train written as $f(x) = \sum_n f_n(x - nL)$, where $f_n(x) \approx 0$ unless $|x| \ll L$. Taking the Fourier transform of this spike-train yields:

$$\tilde{f}(k) = \sum_n e^{in k L} \tilde{f}_n(k) \quad (19)$$

[00155] Since the $e^{in k L}$ term is periodic with period $2\pi/L$ (or fractions thereof), displacement by $2\pi/L$ gives $|\tilde{f}(k + 2\pi/L) - \tilde{f}(k)| = \sum_n |\tilde{f}_n(k + 2\pi/L) - \tilde{f}_n(k)|$. Now because each $f_n(x)$ is nonzero only for $|x| \ll L$, $\tilde{f}_n(k)$ is nearly periodic with period $2\pi/L$, and therefore so is $\tilde{f}(k)$.

[00156] Similarly, the Fourier transform of a nearly-periodic function with period L is a spike-train function with period $2\pi/L$. This is the converse to the Fourier transform of a spike-train function with period L being a nearly-periodic function with period $2\pi/L$. To see why, recall that any continuous function can be written as a piecewise Fourier series:

$$f(x) = \sum_m e^{2\pi i m x/L} f_{m,n}, \quad x \in [(n - 1/2)L, (n + 1/2)L] \quad (20)$$

[00157] If $f(x)$ is nearly periodic, then $|f_{m,n} - f_{m,n+1}| \ll |f_{m,n}|$. With a little perturbation theory, one can trade the discrete $f_{m,n}$ for a continuous $f_m(x)$ that is slowly-varying, i.e. $|f_m(y) - f_m(x)| \ll |f_m(x)|$ if $|y - x| \leq L$. The Fourier transform of each summand of Eq. (20) is the convolution of a delta function $\delta(k - m(2\pi/L))$ and $\tilde{f}_m(k)$. The latter is highly concentrated around $|k| = 0$. Thus $\tilde{f}(k)$ is a spike-train function with period $2\pi/L$.

[00158] It follows from the above two points that the Fourier transform of a nearly-periodic spike-train is another nearly-periodic spike-train. A nearly-periodic spike train may be expressed as:

$$f(x) = \sum_{n=-\infty}^{+\infty} F(x - nL, nL) \quad (21)$$

where $F(x, y)$ (replaces $f_n(x - nL)$ above) is a continuous function that is sharply peaked around $x = 0$ ($F(x, y) \approx 0$ unless $|x| \ll L$) and slowly-varying in y ($F(x, y) \approx F(x, y')$ for $|y - y'| \lesssim L$). The Fourier transform is:

$$\tilde{f}(k) = \frac{\sqrt{2\pi}}{L} \sum_n \tilde{F}\left(\frac{2\pi n}{L}, k - \frac{2\pi n}{L}\right) \quad (22)$$

where $\tilde{F}(k_x, k_y)$ is the 2D Fourier transform of $F(x, y)$. It is slowly-varying in k_x and sharply peaked around $k_y = 0$. As an aside, this technique of reducing the 1D Fourier transform of a slowly varying pulse train to a much smaller and more manageable 2D Fourier transform has unrelated applications in frequency-comb simulations, e.g., in extracting radio-frequency (RF) beat-note spectra of micro-combs or synchronously-pumped optical parametric oscillators (OPOs).

[00159] Additional math shows how the Fourier transform properties of spike trains can be used to perform a discrete Fourier transform (DFT) in the optical domain. FIG. 9A, described above, depicts the optical Fourier transform of a point source. A lensless transform is possible if the Fourier plane is in the far field. A lens brings the Fourier plane to the lens's focal plane as in FIG. 9A.

[00160] FIG. 16 illustrates an optical Fourier transform of a shifted source using a lens 904. In the paraxial approximation, the lens 904 imparts a phase shift $\Delta\phi = -k_0\xi^2/2f + \text{const}$. For an emitter array at the left plane, calculating the pattern at the right plane is a straightforward exercise with path integrals; the result is:

$$E'(x') \propto \int E(x) e^{2\pi i x x' / \lambda f} dx \quad (23)$$

This is the optical Fourier transform.

[00161] FIG. 17 shows an optical system 1720 that converts this optical Fourier transform into a discrete Fourier transform. This system 1720 has consider transceivers 1730 and transceivers (pixels) 1740 located at discrete, evenly-spaced points in the image plane and Fourier plane, respectively: $x'_k = \sqrt{\lambda f / N} k$, $x_l = \sqrt{\lambda f / N} l$. A beam splitter 1760 combines pulses from the image-plane transceivers 1730 with light from weight-server emitters 1750, which are also located in an image plane. In this system 1720, Eq. (23) reduces to the canonical form Eq. (14).

[00162] Each transceiver 1740 in the Fourier plane is composed of sub-pixels 1742. There are C' sub-pixels 1742 per transceiver 1740, one for each output channel. The following derivation is for a 1D detector array, but 2D is a straightforward extension. Suppose that the sub-pixels 1742 are evenly spaced with positions $x'_{k;m} = (k + m/C')s$, where $s = \sqrt{\lambda f / N}$. The field from the neuron inputs $x_{l;n}$ is:

$$E'(x'_{k;m})|_{\text{neurons}} \propto \frac{1}{\sqrt{N}} \sum_l e^{2\pi i (k+m/C')l} x_{l;n}$$

The channels are piped in one at a time, so the n^{th} input channel arrives at time-step n . At time-step n , the field from the weight-server emitters 1750 takes the form:

$$E'(x'_{k;m})|_{\text{weights}} = \tilde{K}_{k;mn} \equiv \sum_l e^{2\pi i (k+m/C')l} K_{l;mn}$$

[00163] (Note the subtle difference between $\tilde{K}$ and the Fourier transform Eq. (14).) The convolution kernel K is much smaller than the image. As a result, its Fourier transform is relatively smooth: $\tilde{K}_{k;mn} \approx \tilde{K}_{k+1;mn}$. This means that, for a given sub-pixel index m , the input field from the weight-server emitters 1750 is also slowly varying; therefore $E'(x')|_{\text{weights}}$ is a nearly periodic function in the sense described above. As a result, the output at the weight server $E(x)|_{\text{weights}}$

takes the form of a spike train. In particular, for C' sub-pixels 1742 per pixel 1740, the weight server's outputs C' clusters of K_x , spaced every N pixels, where N and K_x are the sizes of the image and kernel, and $N \gg K_x$.

[00164] In total, the weight server 1750 has $C'K_x$ independent outputs in each time-step, for $CC'K_x$ total. This is the total number of weights for a (1D) kernel, giving enough degrees of freedom to make it work. There is a Fourier-series relation between the weight-server outputs and the desired weights (e.g., Eq. 19), but this can be pre-computed.

[00165] After C time steps, the (conjugated) heterodyne output from Eqs. (24) and (25) gives the quantity:

$$\sum_{l;n} e^{-2\pi i(k+m/C')l} (\hat{K}_{k;mn}^* x_{l;n})^* \quad (26)$$

[00166] This has an extra m/C' term. This extra term can be eliminated by performing a proper inverse Fourier transform. Recall that in step 2 of the optical Fourier transform (shown in FIG. 12C), the pixels 1740 in the Fourier plane fire and the result is read out by the image-plane transceivers 1730. This time, each channel fires one at a time, from each sub-pixel 1742. So, on time-step m , all of the sub-pixels $x'_{k;m}$ fire, producing the following output at the image plane:

$$\begin{aligned} E(x_j) &\propto \sum_{kl;n} e^{2\pi i(k+m/C')j} e^{-2\pi i(k+m/C')l} (\hat{K}_{k;mn}^* x_{l;n})^* \\ &\propto \sum_{klp;n} e^{2\pi i(k+m/C')j} e^{-2\pi i(k+m/C')l} e^{2\pi i(k+m/C')p} (K_{p;mn}^* x_{l;n})^* \\ &\propto (\sum_{p;n} K_{p;mn}^* x_{j+p;n})^* \end{aligned} \quad (27)$$

The final step of Eq. (27) uses the identity

$$\sum_{k=0}^{N-1} e^{2\pi i(k+\xi)l/N} = N\delta_{l,0} \quad (28)$$

which holds for all ξ (the case $\xi = 0$ is the traditional DFT).

[00167] Recall that in the original FC optical neural network, the neurons fired sequentially, and their signals were broadcast to the neurons of the next layer and then weighted. In the present scheme, each channel fires sequentially, but the neurons fire simultaneously. In the Fourier plane, within each pixel 1740, the channel output n at wave-vector k is likewise broadcast to all m sub-pixels 1742, corresponding to the output channels. C steps implement a $C' \times C$ FC matrix-vector multiplication at every point in k -space. Looking inside each pixel 1740, the operational principle of this CONV unit is the same as the original FC unit. The tricky part was to decorate the input

and output with Fourier optics to convert this FC problem (technically block-diagonal FC, one block for each pixel 1740) into a convolutional problem.

[00168] TABLE 7 summarizes the three channel-encoding schemes from this section. Time-encoding is the simplest but is slowest and requires a large number of reads or writes, which may lead to signal degradation in analog and/or higher energy costs. Frequency encoding is an elegant solution, but requires many modes and WDM elements, and suffers from chromatic aberration in the Fourier transform (which is intrinsically chromatic), limiting its bandwidth. Spatial encoding achieves the same performance as frequency encoding but uses multiple sub-pixel detectors (and transmitters) per pixel. But it has no other apparent defects, so it may be the way to go. Hybrids are also possible.

	Time steps	Complexity	
Time	(Sec. 3.4)	CC'	Multiple reads / writes per (analog) memory.
Frequency	(Sec. 3.5)	C'	C WDM channels. Separate WDM per pixel.
Space	(Sec. 3.6)	C	C' sub-pixels per pixel.

TABLE 7: Different channel-encoding methods

4. Realizing Optical CNNs without Fourier Optics

[00169] Many deep neural networks make extensive use of convolutional layers. The original homodyne-optical approach was designed for fully-connected networks (Sec. 1) and thus not well-suited to the convolutional case; however, a clever use of Fourier optics could realize the convolution with optically-encoded weights (Sec. 3). However, that scheme involves simultaneous measurement of both quadratures (heterodyne detection), as well as (possibly) co-integration of sources and detectors. Finally, the math of the spatial channel multiplexing (sub-pixels) was quite involved. An easier scheme implemented the digital neural network with optical fan-out via Fourier holography (Sec. 2). However, the phases of Fourier holograms are generally scrambled, making coherent detection problematic in this digital optical neural network. Fortunately, it is possible to realize an optical CNN in a coherent mode without the Fourier optics of Sec. 3 using many of the optical fan-out components from Sec. 2. In this type of optical CNN, Fourier optics can be used to fan out optical-domain data but are not used for the convolution itself.

Row-Column Method

[00170] FIG. 18 illustrates a row-column method 1800 for computing a convolution. The image is zero-padded on the right and bottom with $K_x - 1$ and $K_y - 1$ entries so that any 2D convolution (with shifts $\Delta x, \Delta y$) on the image is equivalent to a 1D convolution on the padded, serialized array (with shift $\Delta x + W\Delta y$ if the array is C-ordered).

[00171] The input is an array of dimension $W \times H \times C$ (1802). The channels are sent in one at a time over a total of C time steps. In time step l , the l^{th} channel is zero-padded and serialized (1804) and transmitted as a vertical array of beams (e.g., using Rui's fan-out circuit with fast modulators, arranging the couplers on a line). This signal passes through an optical fan-out circuit like the one shown in FIG. 9A. The kernel to this fan-out has C' blocks (one per output channel), each with $K_x K_y$ points. These points are equally spaced in x , and their y displacements correspond to the different convolution shifts.

[00172] The kernel is a $K_x \times K_y \times C' \times C$ array (1812). Like the input (1802), the kernel is loop over the input channel index over C time steps. In time step l , the kernel of the l^{th} channel (dimension $K_x \times K_y \times C'$) is sent in as a horizontal line of pixels (1814), from a similar device (e.g., grating array or edge coupler array). The line is divided into C' blocks, each of size $K_x K_y$. This is fanned out in the vertical direction giving approximately $(W + K_x)(H + K_y)$ copies of the l^{th} channel of the kernel, one on each row (1816).

[00173] Finally, the kernel and image outputs are combined onto a photodetector array (1820). Since the photodetector array only needs to collect light (unlike the photodetectors coupled to logic in Sec. 3), this photodetector array can be very slow, simple, compact, cheap, and high-resolution. Reading out the photodetector array twice with opposite phases makes it possible to compute the homodyne signal, which gives the product between the fields (assuming the phases are controlled for).

[00174] The photodetector array has dimension $(K_x K_y C') \times (W' H')$. Each row corresponds to a pair of image coordinates $y \leftrightarrow (i, j)$ (there are $W' H'$ rows for an image of size $W' \times H'$). The columns are grouped into C' blocks for the channels, and each block containing $K_x K_y$ columns which can be indexed by coordinates $(i' < K_x, j' < K_y)$. Thus, each point on the grid can be uniquely assigned five indices (i, i', j, j', k) . The homodyne signal is:

$$z_{ij,i'j',k} = K_{ij',kl} x_{i+i',j+j',l} \quad (29)$$

[00175] Read-out circuitry integrates the photocurrent over all C input channels. This sums over l . Next, within each block, the photocurrents in each row are summed (light-blue box in figure). This sums over (i', j') . At the end, one obtains:

$$y_{ij,k} = \sum_{i'j'l} z_{ij,i'j',k} = \sum_{i'j'l} K_{i'j',kl} x_{i+i',j+j',l} \quad (30)$$

This is exactly the desired convolution function.

[00176] One potential problem with this scheme is achieving the desired phase relation between image and kernel. Recall that the image data is fanned out with Fourier holography. The fan-out kernel for the Fourier hologram includes an array of dots with equal amplitude and different phases. Since there is one dot per column, the phase is column-dependent and is denoted ϕ_x (see FIG. 18). The same problem occurs with the kernel fan-out. Here the fan-out is in the vertical direction, and each dot on the fan-out kernel also has its own phase, denoted ϕ_y . Without any phase correction, the actual photocurrent is $y_{ij,k} = \sum_{i'j'l} \text{Re} [K_{i'j',kl} x_{i+i',j+j',l} e^{i(\phi_x - \phi_y)}]$. Since the kernel and image are real-valued, it might be possible to compensate this by rescaling the photocurrents by $\cos(\phi_x - \phi_y)^{-1}$; however, this would involve lots of extra computation and could be problematic if $\phi_x - \phi_y \approx \pi/2$ and the scaling factor is very large.

[00177] Another way to fix this problem is to correct the phase ϕ_x optically using an optical phase shifter before each grating coupler to apply a phase $\psi_x = \phi_x$ to the kernel outputs (FIG. 18). Applying a 2D phase mask $\psi_y = -\phi_y$ to the kernel output after fan-out compensate ϕ_y . The pattern of the dots, and therefore ϕ_x , for the image fan-out depends on the convolution parameters C' , K_x and K_y ; therefore, the phases ψ_x should be tunable. However, the fan-out for the kernel is always the same, so the phase mask ψ_y may be static (fixed) and can be implemented with a printed element or cylindrical lens rather than an SLM.

Displaced Image Method

[00178] FIG. 19 illustrates a displaced image method 1900 for computing a convolution. As before, the goal is to compute $y_{ij,k}$ (dimension $W' \times H' \times C$) from input $x_{ij,l}$ (dimension $W \times H \times C$) (1902). As in the row-column method 1800 in FIG. 18, following Eq. (30), the displaced image method 1900 first iterates over the input channel index l (C channels in all). For each channel, the method 1900 iterates over the $K_x K_y$ displacements performed for the convolution. Each displacement is represented by a unique index pair (i', j') in Eq. (30) (1904).

The displaced image has $W \times H$ pixels, with a focused spot at the center of each pixel. There is enough space between the spots that each pixel can be fanned out to C' identical sub-pixels without overlap (1906). This image is sent to the detector array.

[00179] Simultaneously, for each channel, the displaced image method 1900 iterates over the (i', j') indices of the kernel (1912). For a particular index pair (i', j') , the kernel $K_{i'j',kl}$ has C' elements (denoted by index k , the other indices fixed); these elements are encoded onto the sub-pixels of the center pixel (1914). A Fourier hologram fans this out with equal intensity to all pixels of the image (1916). This kernel is interfered with the image on the detector array, computing the homodyne product (1920).

[00180] The photodetector array has $(W + K_x - 1)(H + K_y - 1)$ pixels, with C' sub-pixel detectors per pixel. Each detector reads out the convolution $y_{ij,k}$, where (i, j) are the (pixel) coordinates and k is the sub-pixel index.

[00181] As before, the fan-out phases ϕ_{ij} and ϕ_k should be corrected so that the signals interfere correctly at the detector array. Fortunately, this can be done by placing phase modulators before every grating coupler at the transmitter chips for the image and kernel. These are denoted in FIG. 19 as ψ_{ij} and ψ_k , respectively. If $\psi_{ij} = \phi_{ij}$ and $\psi_k = \phi_k$, the fan-out phases are compensated for and the detector array computes the product in Eq. (30).

Convolution Processor

[00182] FIG. 20 illustrates a processor 2000 for computing a convolution using the row-column method of FIG. 18 or the displaced image method of FIG. 19. The processor 2000 includes an image transmitter array 2002, like the transmitter array in FIG. 5B, that pads and serializes image data 2001 in the row-column method or displaces image data 2001 in the displaced image method. The image transmitter array 2002 does this by modulating an electrical-domain representation of the image data 2001 onto many copies of an optical pulse train to produce optical image data 2003.

[00183] The processor 2000 includes a first $4f$ optical system 2010 that displaces and fans out the optical image data 2003 in the row-column method or fans out the optical image data 2003 in the displaced image method. More specifically, the first $4f$ optical system 2010 has a first lens 2012 that spatially Fourier transforms the optical image data 2003. A first SLM 2014 in the Fourier

plane (the focal plane of the first lens 2012) spatially phase modulates the Fourier-transformed optical image data; the result is spatially Fourier transformed by a second lens 2016 to produce a desired spatial shift at the output of the first $4f$ optical system 2010.

[00184] A kernel transmitter array 2052 serializes kernel data (weights) 2051 in the row-column method or displaces the kernel data 2051 in the displaced image method. The kernel transmitter array 2052 does this by modulating an electrical-domain representation of the image data 2051 onto many copies of an optical pulse train to produce optical kernel data 2053. This optical pulse train is coherent with the optical pulse train used to drive the image transmitter array 2002. Both optical pulse trains may be produced by the same source, such as a pulsed laser whose output is split by a beam splitter.

[00185] A second $4f$ optical system 2060 displaces and fans out the optical kernel data 2053 in the row-column method or fans out the optical kernel data 2053 in the displaced image method. Like the first $4f$ optical system 2010, the second $4f$ optical system has a first lens 2062 that spatially Fourier transforms the optical kernel data 2053. A second SLM 2064 in the Fourier plane (the focal plane of the first lens 2062) spatially phase modulates the Fourier-transformed optical kernel data, and the result is spatially Fourier transformed by a second lens 2066 to produce a desired spatial shift at the output of the second $4f$ optical system 2060.

[00186] A mirror 2070 reflects the output of the second $4f$ optical system 2060 to a beam splitter 2020 that combines the outputs of the first $4f$ optical system 2010 and the second $4f$ optical system 2060. A detector array 2090 senses combined output, which is a convolution of the image data 2001 and the kernel data 2051. When implementing the row-column method, a 2D phase mask 2084 in an image plane of the output of the second $4f$ optical system 2060 applies a 2D phase modulation $\psi_y = -\phi_y$ to the kernel output after fan-out to compensate for the row-dependent phase ϕ_y . When implementing the displaced image method, this 2D phase mask 2084 may be omitted in favor of phase modulators in the image transmitter array 2002 and kernel transmitter array 2052 that apply fan-out phases ϕ_{ij} and ϕ_k , respectively, the signals interfere correctly at the detector array 2090.

Resource Consumption

[00187] These neural network architectures can be compared by looking at how they perform on certain resource metrics. These metrics include (1) number of time steps, (2) number

of transmitters and detectors, and (3) input- and kernel-fanout. TABLE 8 compares these figures for the Fourier-transform ONN (Sec. 3), row-column method, and displaced image method, and electronics. Since all of these schemes send the input channels in sequentially, the number of time steps is a multiple of C . However, the displaced image method displaces the image $K_x K_y$ times, so it performs $C K_x K_y$ steps, as does the standard electronic process.

	Electronics	FT-ONN (Sec. 3)	RC Method (Sec. 4.1)	DI Method (Sec. 4.2)
Time steps	$\geq C K_x K_y$	$2C$	C	$C K_x K_y$
Input transmitters	–	$2W'H'$	$W'H'$	$W'H'$
Kernel transmitters	–	$C'K_x K_y$	$C'K_x K_y$	$K_x K_y$
Detectors N_{det}	–	$2W'H'C'$	$W'H'C'K_x K_y$	$W'H'C'$
Detectors/output	–	2	$K_x K_y$	1
Input reads	$W'H'C'CK_x K_y$	$2W'H'C$	$W'H'C$	$W'H'CK_x K_y$
Input fan-out F_{in}	–	$1/2 C'K_x K_y^\dagger$	$C'K_x K_y$	C'
Kernel reads	$W'H'C'CK_x K_y$	$2C'CK_x K_y$	$C'CK_x K_y$	$C'CK_x K_y$
Kernel fan-out F_{ker}	–	$1/2 W'H'^\dagger$	$W'H'$	$W'H'$
Complexity	–	Optical FT	Extra detectors	Image shifting

TABLE 8: Resource consumption of different neural network architectures including electronics, the Fourier-Transform optical neural network (FT-ONN), the Row-Column (RC) method, and the Displaced Image (DI) method. Assuming stride of 1 with $W' = W, H = H$. † FT-ONN has 2 steps with fan-out $C'K_x K_y$ per step.

[00188] TABLE 8 shows that there is a tradeoff between spatial and temporal complexity. The row-column method uses C time steps, but it uses $K_x K_y$ detectors per output, while the displaced image method uses one detector per output. The product between the number of time-steps and the number of detectors is conserved ($N_{\text{step}} N_{\text{det}} = W'H'C'CK_x K_y$), consistent with the fact that both of these networks are performing the convolution in the conventional way, which involves $W'H'C'CK_x K_y$ MACs. Without wavelength multiplexing, each detector performs one MAC per time step. However, with the aid of a Fourier transform, the convolution uses just $W'H'C'C$ MACs plus three FT's. As a result, the FT-ONN is more efficient than the row-column

and displaced image methods with $N_{\text{step}}N_{\text{det}} = 2W'H'C'C$ (the factor of 2 comes from the need for detectors at both image plane and Fourier plane). However, the FT-ONN also uses an optical Fourier transform.

[00189] Detector count is significant because the pixel count of the camera limits the size of images used in the row-column method much more severely than the displaced image method, especially for large kernels, and because each pixel has finite size and thus finite capacitance. Suppose each detector uses an energy E_{det} to get a reasonable readout SNR. In the displaced image method, the optical energy per output is thus $\geq E_{\text{det}}$. But in the row-column method, each output is the average of $K_x K_y$ detectors, giving an energy $\geq K_x K_y E_{\text{det}}$. This may be significant depending on how large E_{det} is. The optical energy bound per MAC for sufficient detector charging will be:

$$E_{\text{opt}} = \frac{N_{\text{det}}}{W'H'C'CK_x K_y} E_{\text{det}} \quad (31)$$

[00190] Another factor is input- and kernel-fanout (F_{in} , F_{ker}). Since memory reads dominate energy consumption in digital neural networks, a significant savings can be obtained by reading the memory once and fanning the result out to many detectors. Let E_{tr} be the total transmitter energy, including the memory read, conversion to analog, and the electrical cost of driving a modulator. Then the electrical energy per MAC is:

$$E_{\text{el}} = \left(\frac{1}{F_{\text{in}}} + \frac{1}{F_{\text{ker}}} \right) E_{\text{tr}} \quad (32)$$

The total energy per MAC is $E_{\text{mac}} = E_{\text{el}} + \eta^{-1} E_{\text{opt}}$, where η is the product of various efficiencies—detector, modulator, lightsource, etc.

Conclusion

[00191] While various inventive embodiments have been described and illustrated herein, those of ordinary skill in the art will readily envision a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the inventive embodiments described herein. More generally, those skilled in the art will readily appreciate that all parameters, dimensions, materials, and configurations described herein are meant to be exemplary and that the actual parameters, dimensions, materials, and/or configurations will depend upon the specific application or applications for which the inventive teachings is/are used. Those skilled in the art will recognize or be able to ascertain, using no more than routine

experimentation, many equivalents to the specific inventive embodiments described herein. It is, therefore, to be understood that the foregoing embodiments are presented by way of example only and that, within the scope of the appended claims and equivalents thereto, inventive embodiments may be practiced otherwise than as specifically described and claimed. Inventive embodiments of the present disclosure are directed to each individual feature, system, article, material, kit, and/or method described herein. In addition, any combination of two or more such features, systems, articles, materials, kits, and/or methods, if such features, systems, articles, materials, kits, and/or methods are not mutually inconsistent, is included within the inventive scope of the present disclosure.

[00192] Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[00193] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[00194] The indefinite articles “a” and “an,” as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean “at least one.”

[00195] The phrase “and/or,” as used herein in the specification and in the claims, should be understood to mean “either or both” of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with “and/or” should be construed in the same fashion, i.e., “one or more” of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the “and/or” clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to “A and/or B”, when used in conjunction with open-ended language such as “comprising” can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

[00196] As used herein in the specification and in the claims, “or” should be understood to have the same meaning as “and/or” as defined above. For example, when separating items in a list, “or” or “and/or” shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as “only one of” or “exactly one of,” or, when used in the claims, “consisting of,” will refer to the inclusion of exactly one element of a number or list of elements. In general, the term “or” as used herein shall only be interpreted as indicating exclusive alternatives (i.e., “one or the other but not both”) when preceded by terms of exclusivity, such as “either,” “one of,” “only one of,” or “exactly one of.” “Consisting essentially of,” when used in the claims, shall have its ordinary meaning as used in the field of patent law.

[00197] As used herein in the specification and in the claims, the phrase “at least one,” in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase “at least one” refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, “at least one of A and B” (or, equivalently, “at least one of A or B,” or, equivalently “at least one of A and/or B”) can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[00198] In the claims, as well as in the specification above, all transitional phrases such as “comprising,” “including,” “carrying,” “having,” “containing,” “involving,” “holding,” “composed of,” and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases “consisting of” and “consisting essentially of” shall be closed or semi-closed transitional phrases, respectively, as set forth in the United States Patent Office Manual of Patent Examining Procedures, Section 2111.03.

CLAIMS

1. An optical neural network comprising:
 - a first optical fan-out element to produce M copies of N optical weight signals;
 - a second optical fan-out element to create N copies of M optical input signals;
 - a two-dimensional array of homodyne receivers, in optical communication with the first optical fan-out element and the second optical fan-out element, to produce electronic signals representing interference of each of the N copies of the M optical input signals with a corresponding copy of the M copies of the N optical weight signals;
 - electronic circuitry, operably coupled to the two-dimensional array of homodyne receivers, to apply a nonlinear function to the electronic signals; and
 - a light source, operably coupled to the electronic circuitry, to emit an optical output signal representing a matrix multiplication of the M optical input signals with the N optical weight signals,wherein M and N are positive integers.
2. The optical neural network of claim 1, wherein the first optical fan-out element, the second optical fan-out element, the two-dimensional array of homodyne receivers, the electronic circuitry, and the light source form a convolutional layer in a series of layers in the optical neural network.
3. The optical neural network of claim 2, wherein the series of layers comprises additional convolutional layers and at least one fully connected layer.
4. The optical neural network in claim 2, further comprising:
 - another series of layers to compute a matrix multiplication of the N optical weight signals with another M optical input signals.
5. The optical neural network of claim 1, wherein N is about 1,000 to about 10,000,000.
6. The optical neural network of claim 1, wherein the N copies of the M optical input signals propagate in free space between the second fan-out element and the array of homodyne receivers.

7. The optical neural network of claim 1, wherein the optical input signal encodes a vector length N and each corresponding optical weight signal represents a row of a weight matrix.
8. The optical neural network of claim 1, wherein each homodyne receivers in the array of homodyne receivers comprises:
 - a two-port beam splitter to interfere the copy of the optical input signal and the corresponding optical weight signal; and
 - a differential detector, in optical communication with the two-port beam splitter, to detect light emitted by outputs of the two-port beam splitter.
9. The optical neural network of claim 1, wherein the electronic circuitry comprises:
 - an array of analog-to-digital converters (ADCs), each ADC in the array of ADCs operably coupled to a corresponding homodyne receiver in the array of homodyne receivers comprises and configured to digitize the electronic signal emitted by the corresponding homodyne receiver; and
 - digital logic, operably coupled to the array of ADCs, to apply the nonlinear function to the electronic signals from the array of ADCs; and
 - an array of digital-to-analog converters (DACs), operably coupled to the digital logic and the light source, to convert the electronic signals into analog signals for modulating the light source.
10. The optical neural network in claim 1, further comprising:
 - an array of optical modulators to modulate copies of an optical pulse train with weights representing a corresponding row in a weight matrix to produce the N optical weight signals.
11. The optical neural network in claim 1, further comprising:
 - a beam splitter, in optical communication with the first optical fan-out element and the second optical fan-out element, to combine each of the N copies of the M optical input signals with the corresponding copy of the M copies of the N optical weight signals.
12. A method of computing an output of a layer of an optical neural network, the comprising:
 - producing N optical weight signals, where N is a positive integer;
 - producing M copies of the N optical weight signals, where M is a positive integer;

creating N copies of M optical input signals;
transducing, with an array of homodyne receivers, interference of each of the N copies of the M optical input signals with a corresponding copy of the M copies of the N optical weight signals into electronic signals;
applying a nonlinear function to the electronic signals; and
emitting an optical output signal representing a matrix multiplication of the M optical input signals with the N optical weight signals.

13. The method of claim 12, wherein the layer is a convolutional layer in the optical neural network.

14. The method of claim 12, further comprising:
computing a matrix multiplication of the N optical weight signals with another M optical input signals.

15. The method of claim 12, wherein the N copies of the M optical input signals propagate in free space to the array of homodyne receivers.

16. The method of claim 12, wherein the optical input signal encodes a vector length N and each corresponding optical weight signal represents a row of a weight matrix.

17. A convolutional layer for a coherent optical neural network, the convolutional layer comprising:
an image-plane transceiver array to emit an array of input signals;
a weight server transmitter array to emit an array of weight signals;
a beam splitter, in optical communication with the image-plane transceiver array and the weight server transmitter array, to combine the array of input signals with the array of weight signals; and
a Fourier-plane transceiver array, in a Fourier plane of the image-plane transceiver array and in optical communication with the beam splitter, to detect a homodyne product of a spatial Fourier transform of the array of input signals and a spatial Fourier transform of the array of weight signals.

18. The convolutional layer of claim 17, wherein the Fourier-plane transceiver array is configured to emit an array of product signals representing the homodyne product of the spatial Fourier transform of the array of input signals and the spatial Fourier transform of the array of weight signals and the image-plane transceiver array is configured to coherently detect a spatial Fourier transform of the array of product signals.

19. The convolutional layer of claim 17, wherein each transceiver in the Fourier-plane transceiver array is configured to detect an in-phase component and a quadrature component of the product of the spatial Fourier transform of the array of input signals and the spatial Fourier transform of the array of weight signals.

20. The convolutional layer of claim 17, wherein each transceiver in the Fourier-plane transceiver array comprises at least one detector element per output channel of the convolutional layer.

21. A method for fanning out data in a digital optical neural network, the method comprising:
for each image in a set of images, the set of images representing an input to a layer in the digital optical neural network,

 breaking the image into a set of patches, each patch in the set of patches
 comprising $K_x \times K_y$ pixels; and

 for each patch in the set of patches,

 spatially convolving the patch with a box function;

 spatially convolving a corresponding kernel with a size of $K_x \times K_y$ with a
 lattice function having a horizontal lattice spacing of K_x and a vertical lattice
 spacing of K_y ; and

 imaging the patch and the corresponding kernel onto a detector array.

FIG. 1

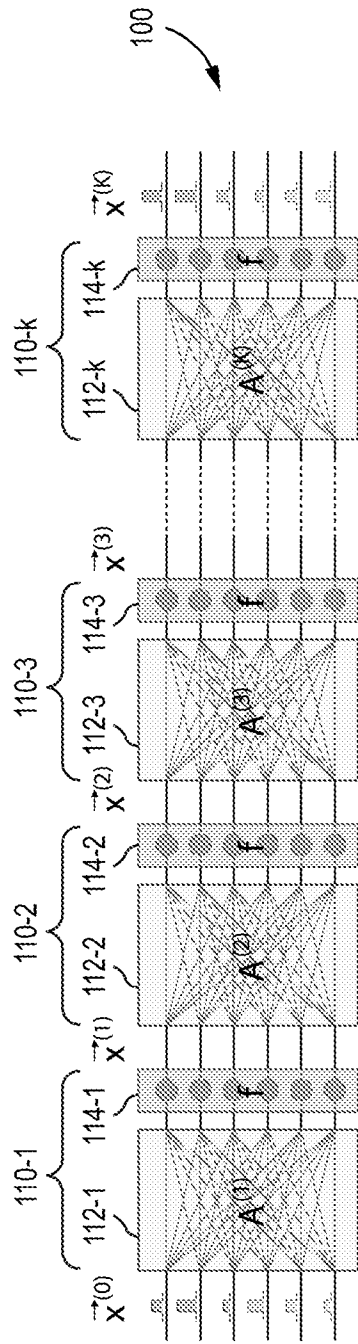
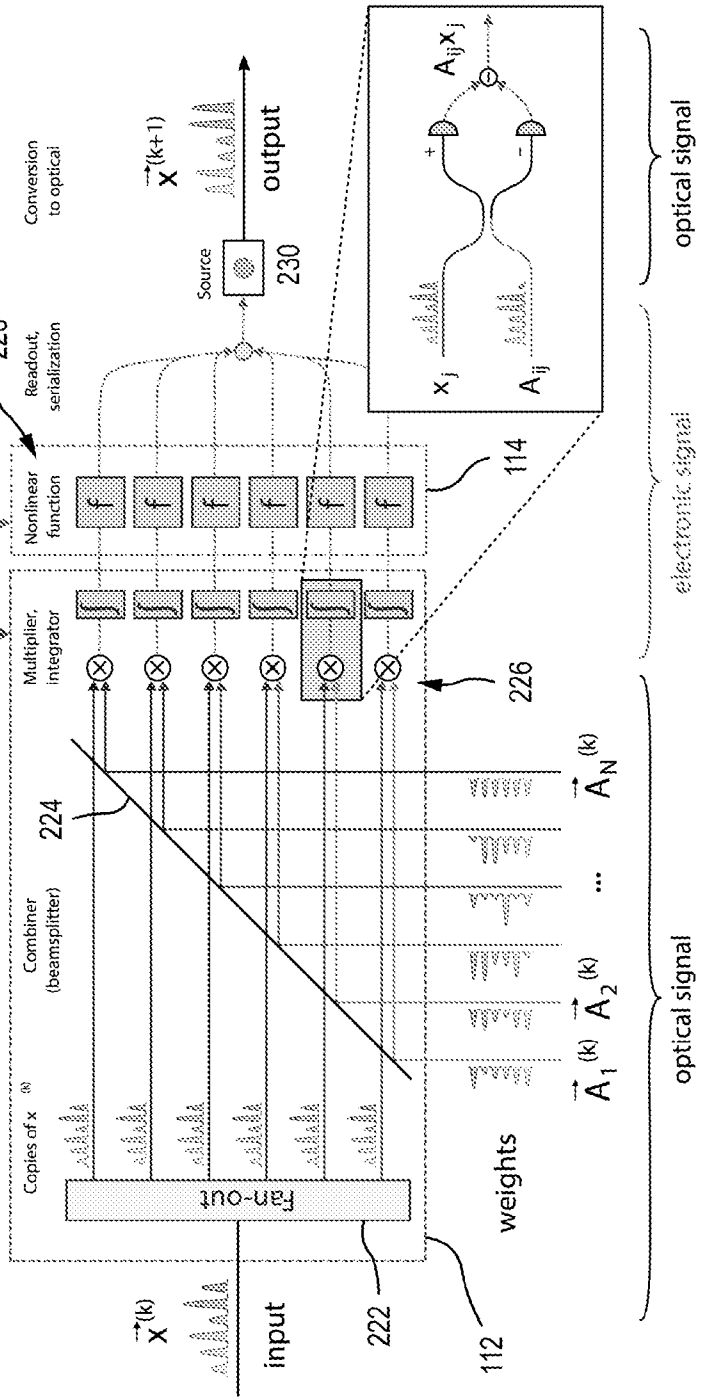


FIG. 2



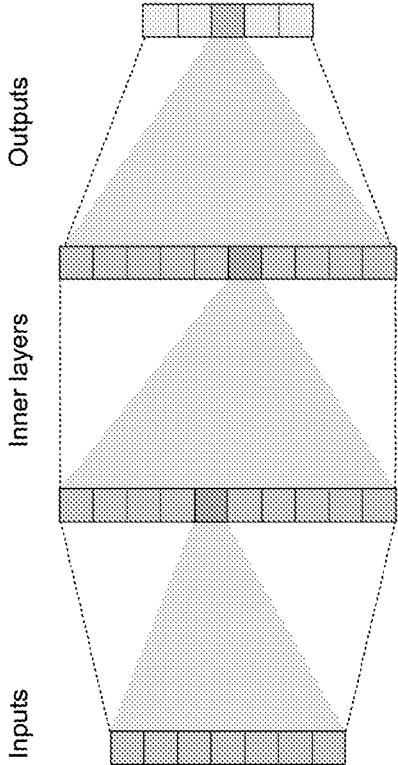


FIG. 3A

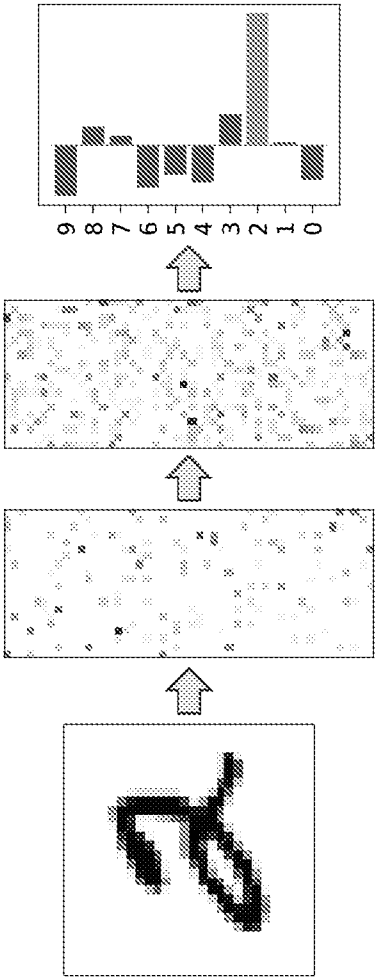


FIG. 3B

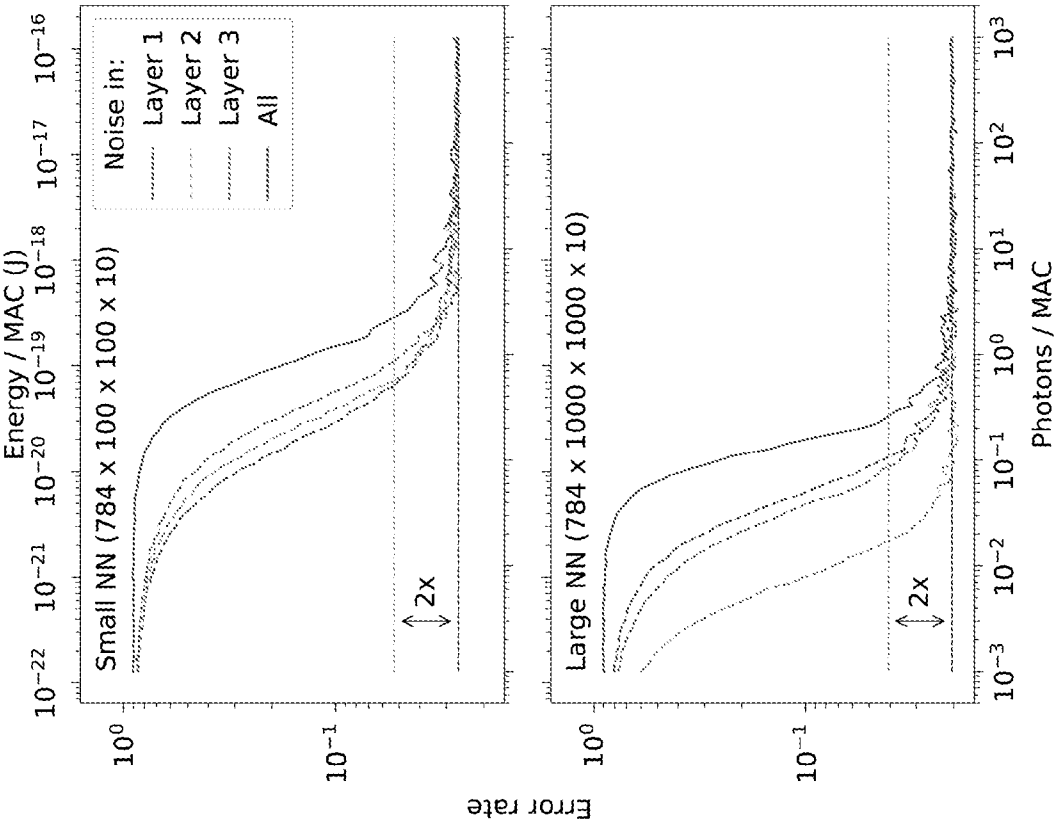


FIG. 3C

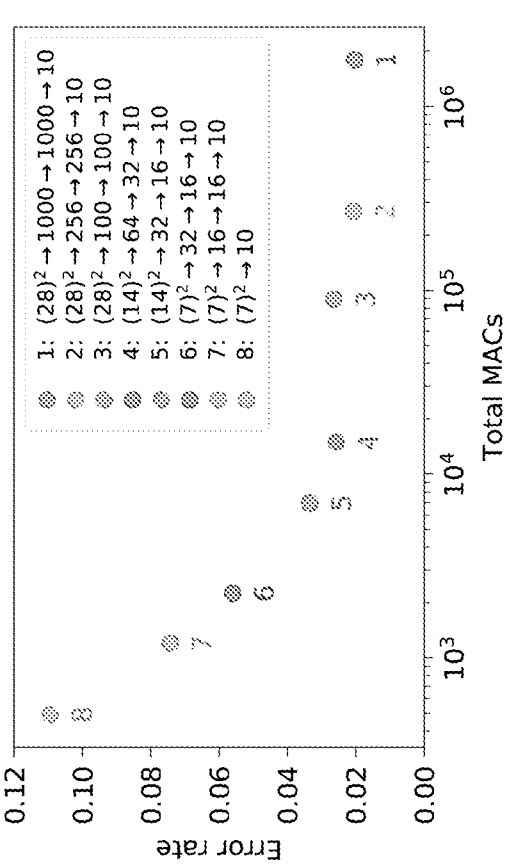


FIG. 4A

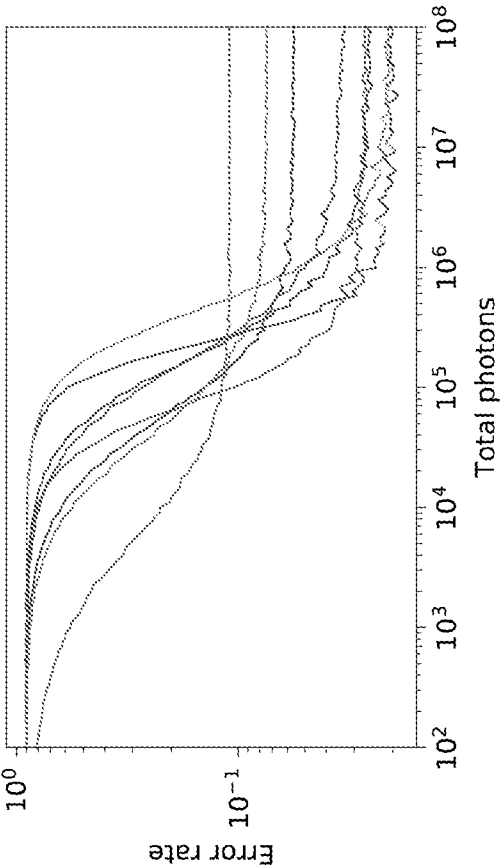


FIG. 4B

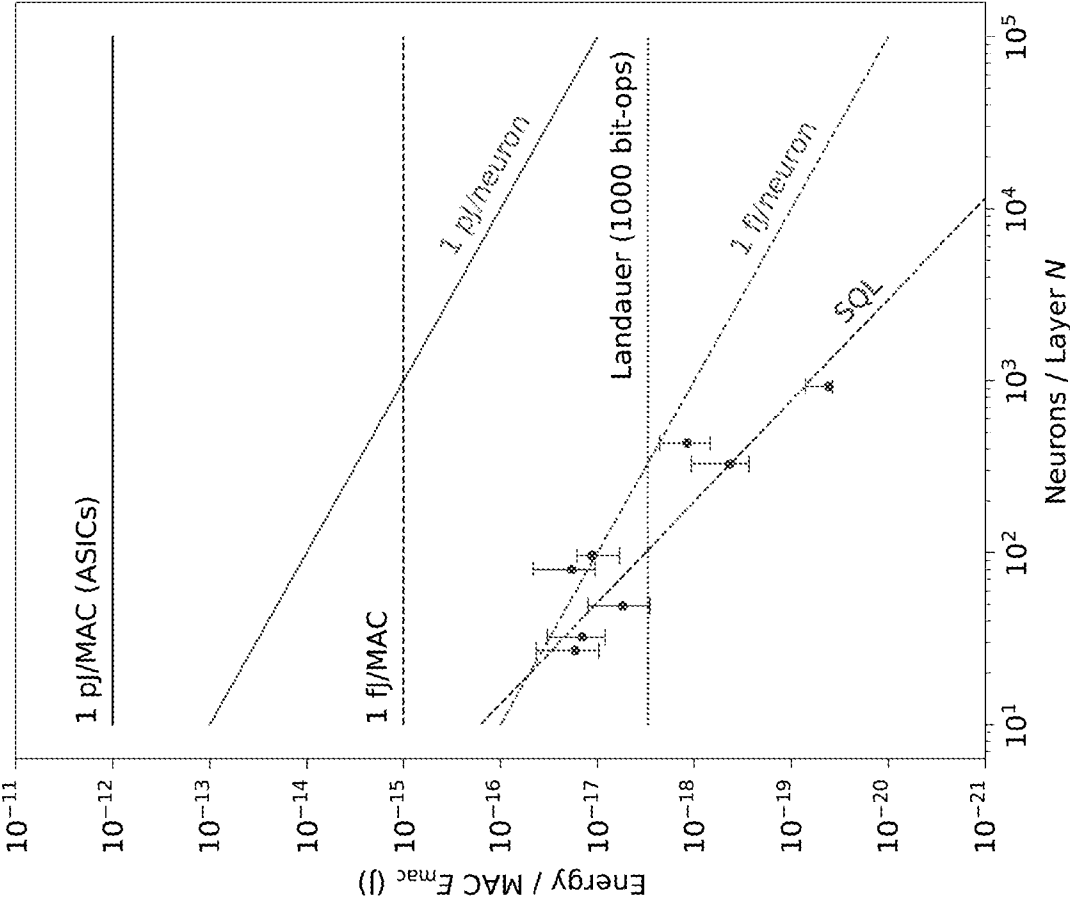


FIG. 4C

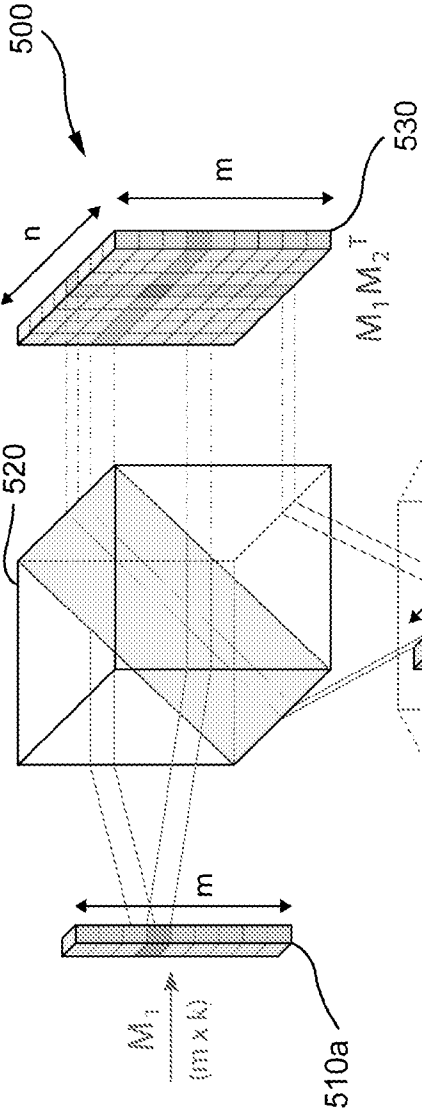


FIG. 5A

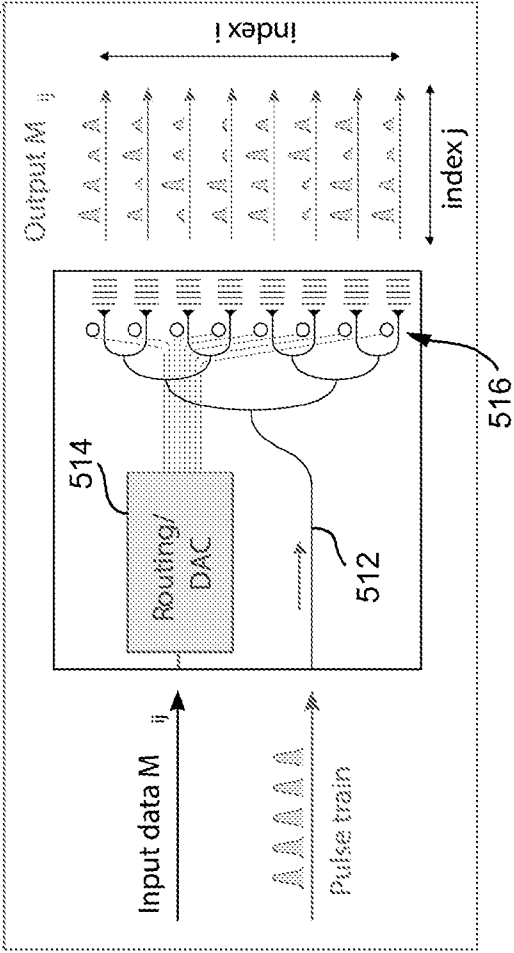


FIG. 5B

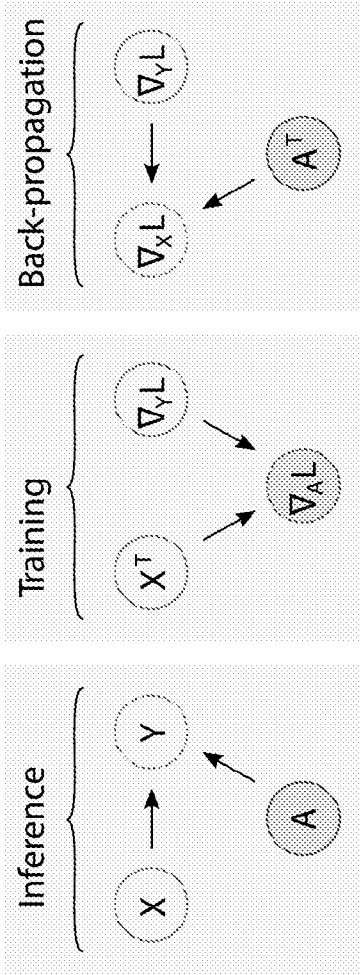


FIG. 5C

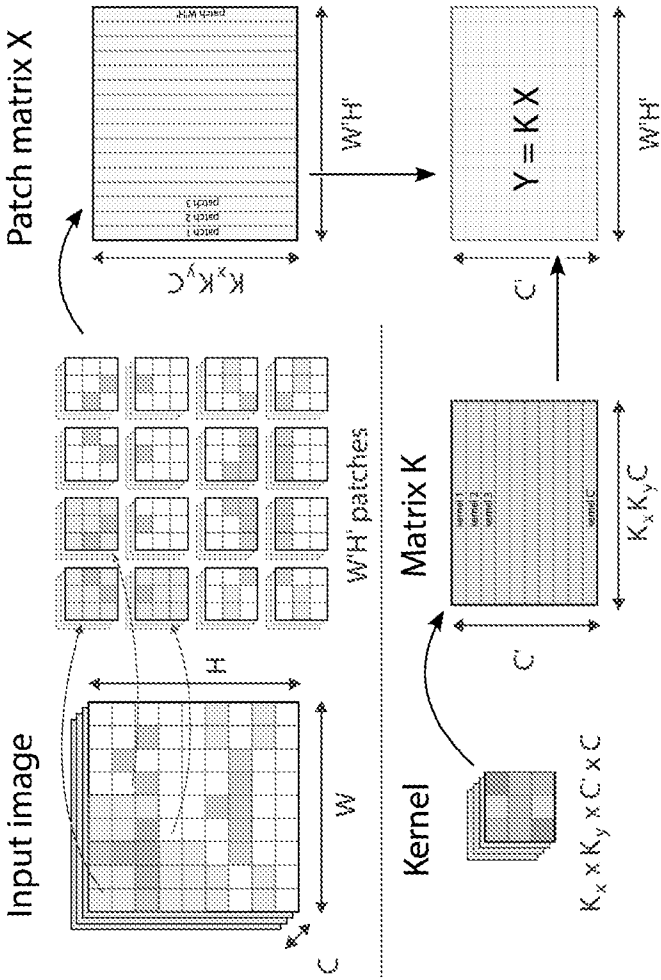


FIG. 5D

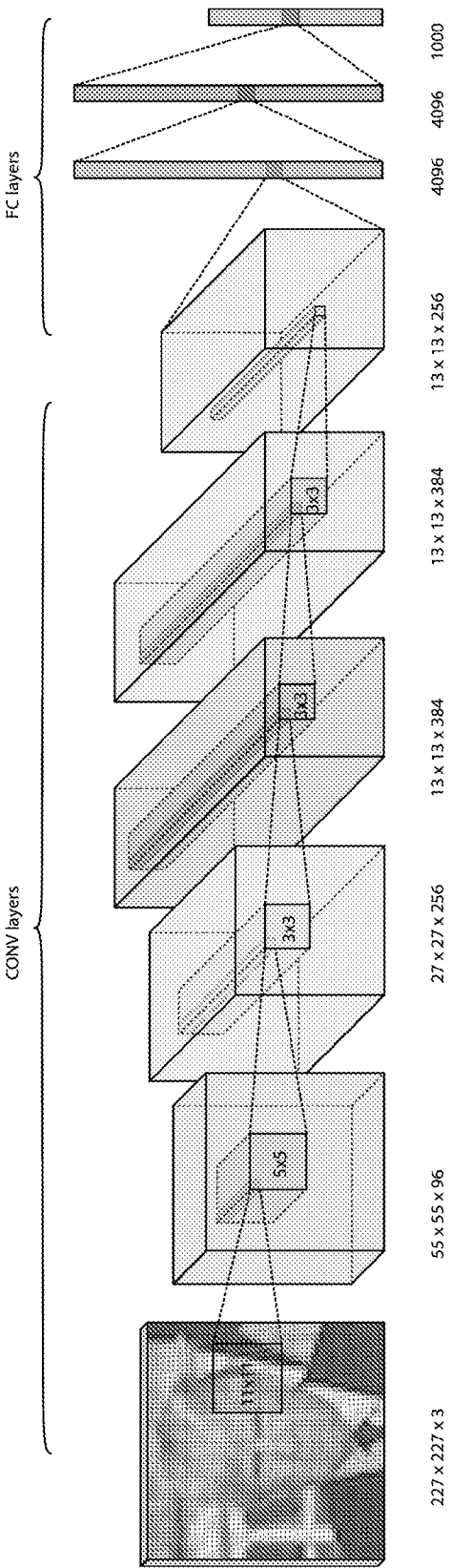


FIG. 6A

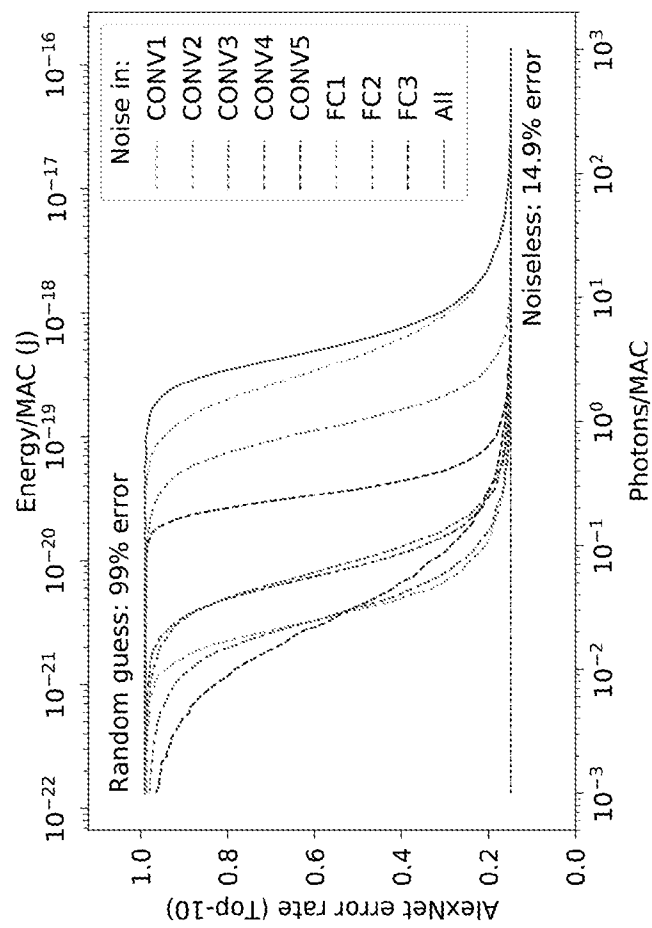


FIG. 6B

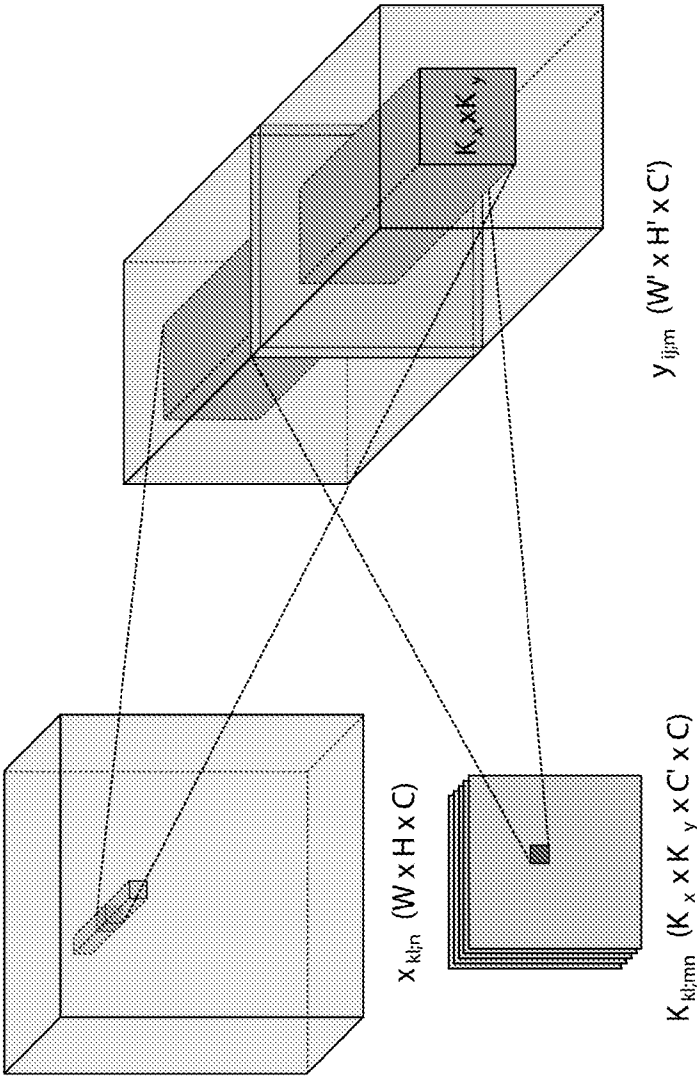


FIG. 7A

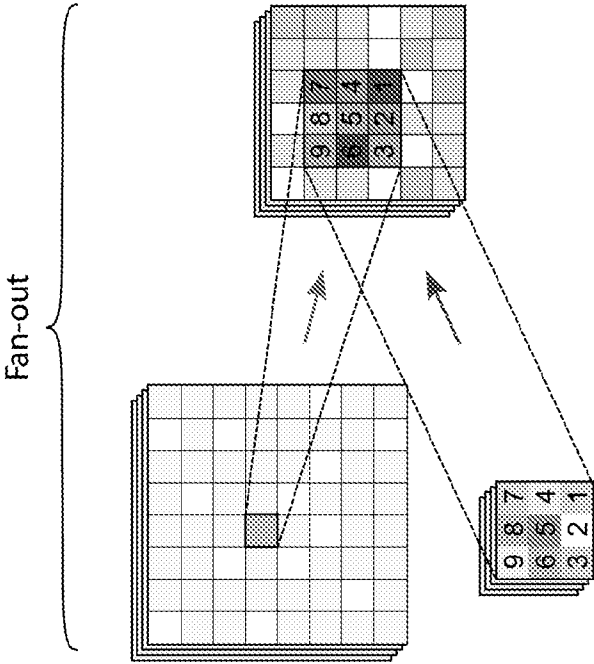


FIG. 7C

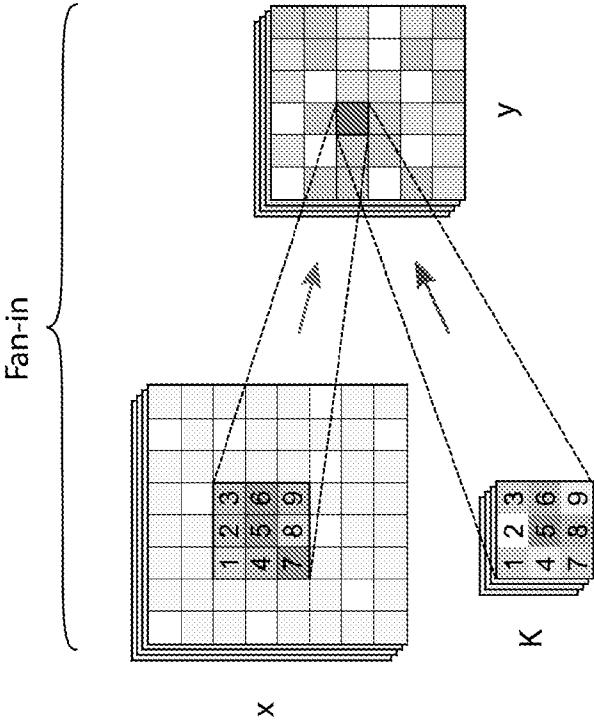


FIG. 7B

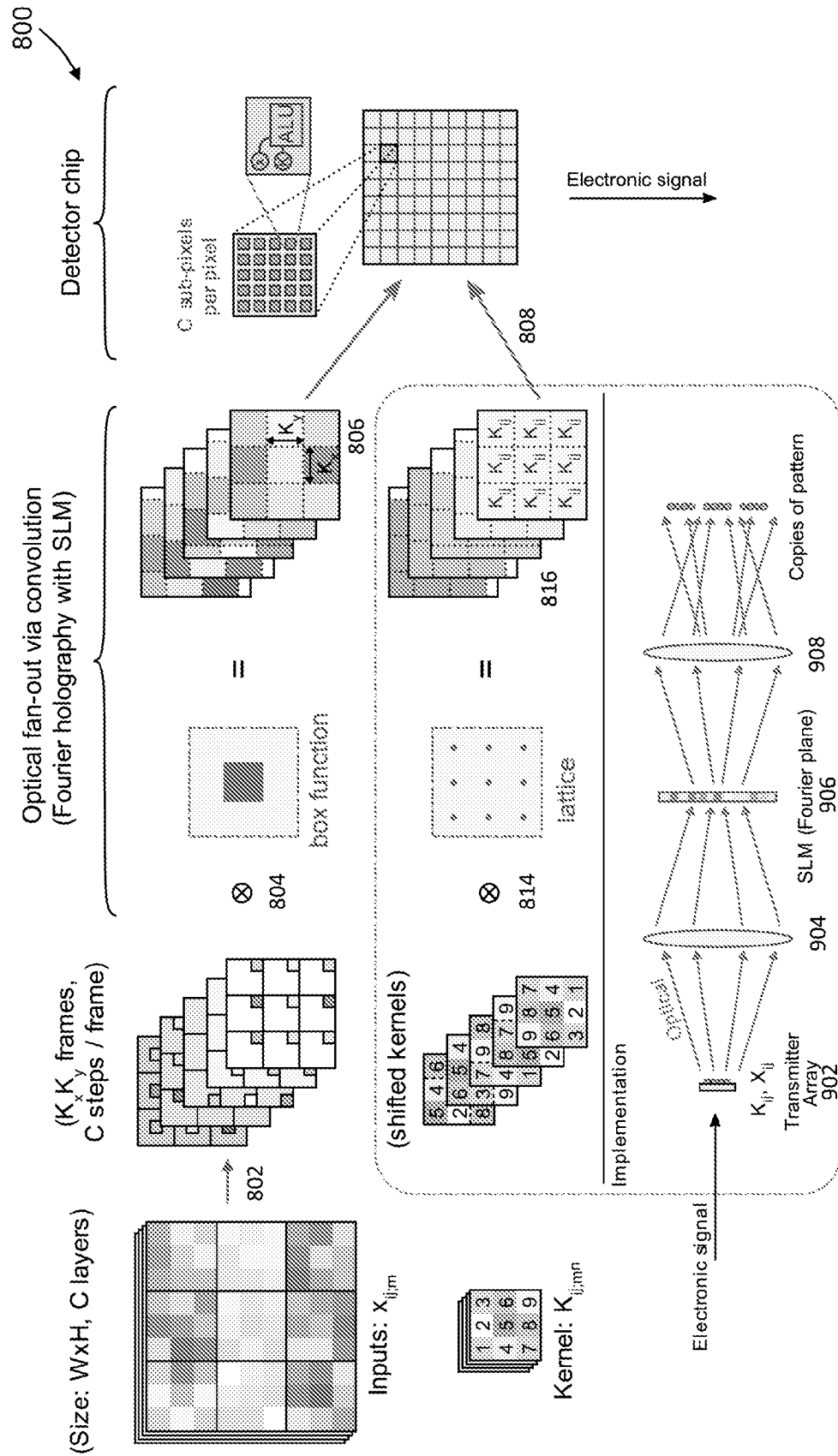


FIG. 8

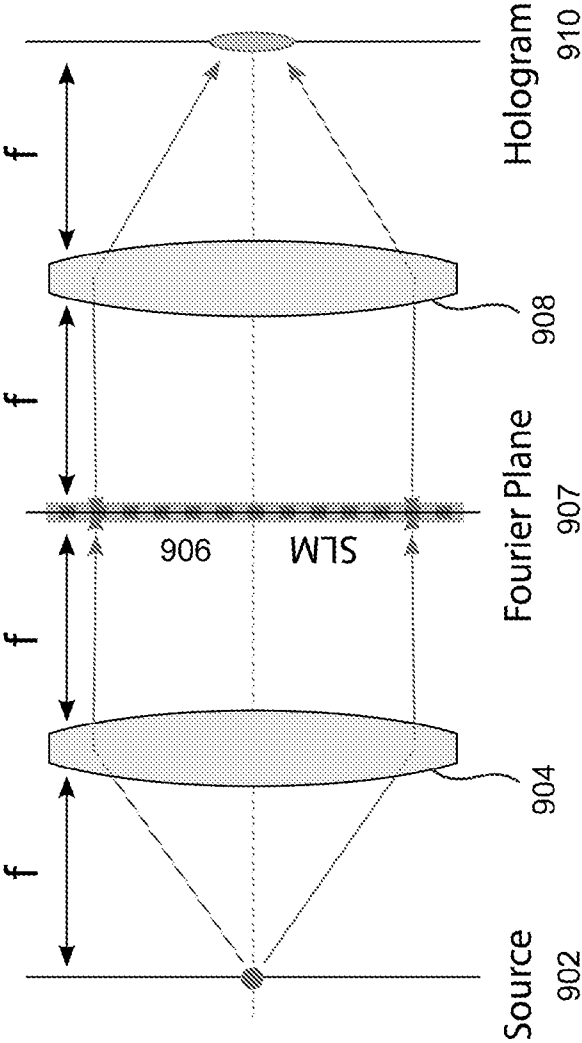


FIG. 9A

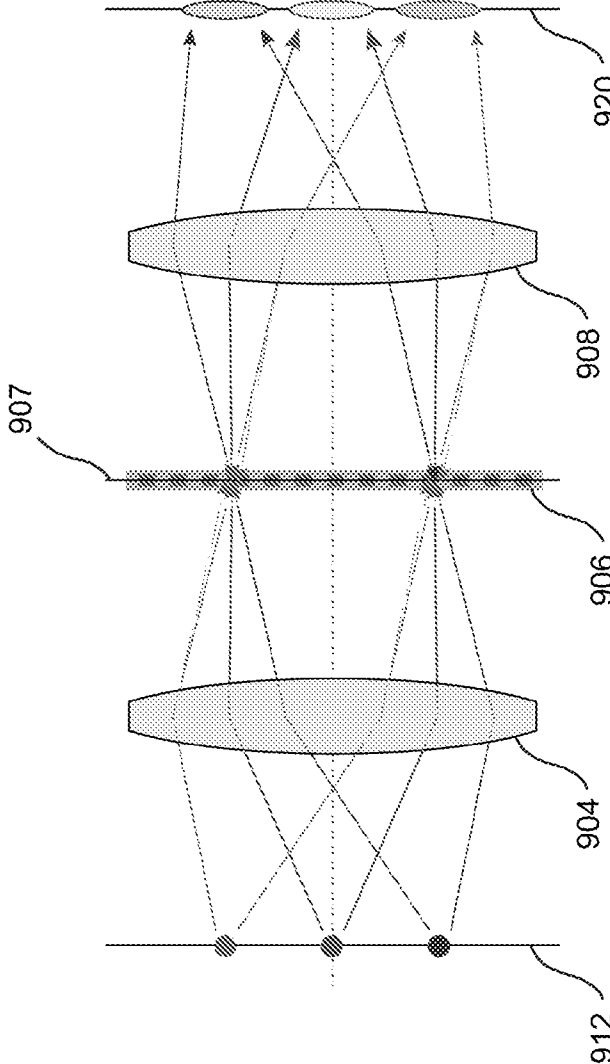


FIG. 9B

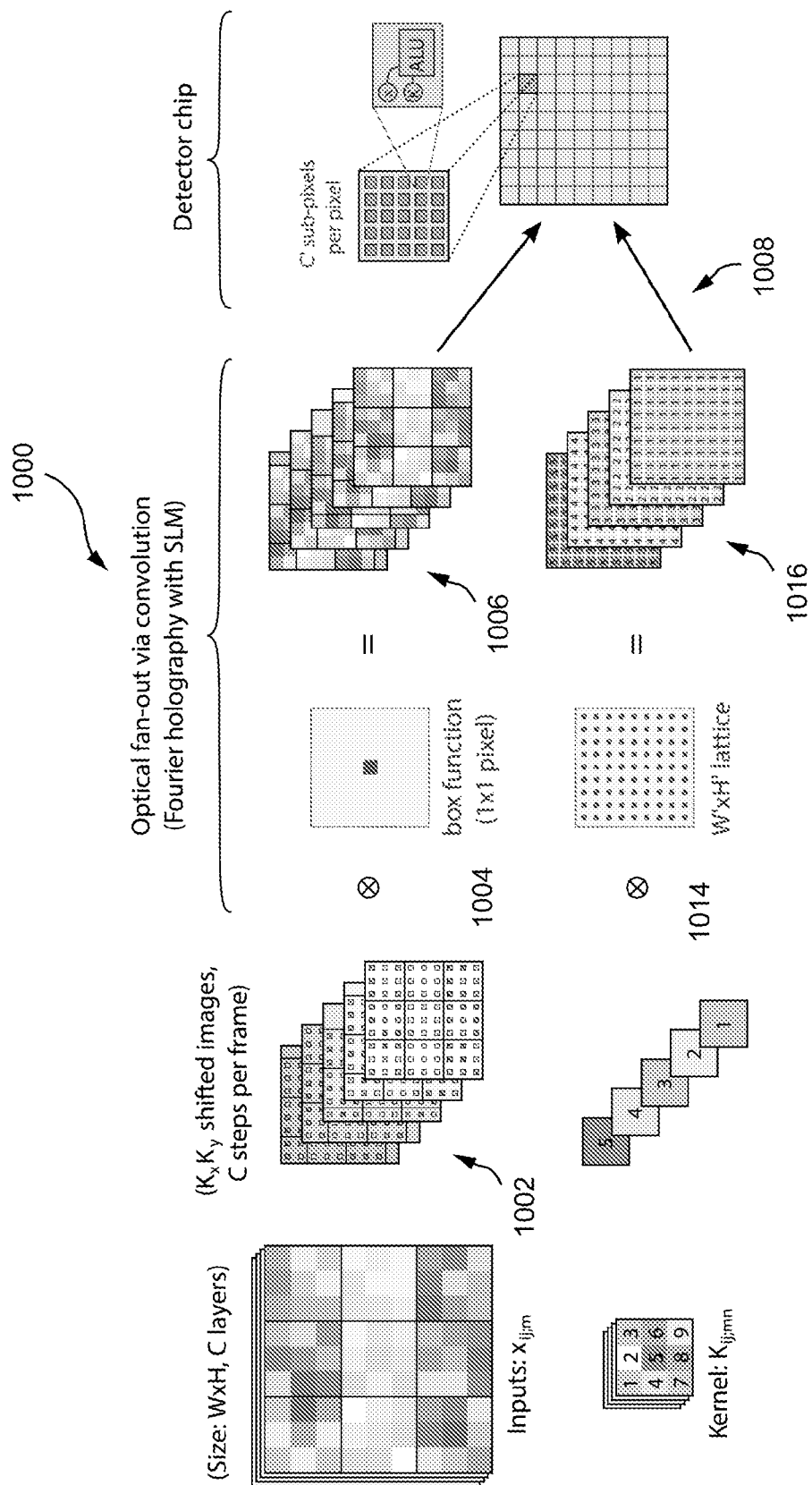


FIG. 10

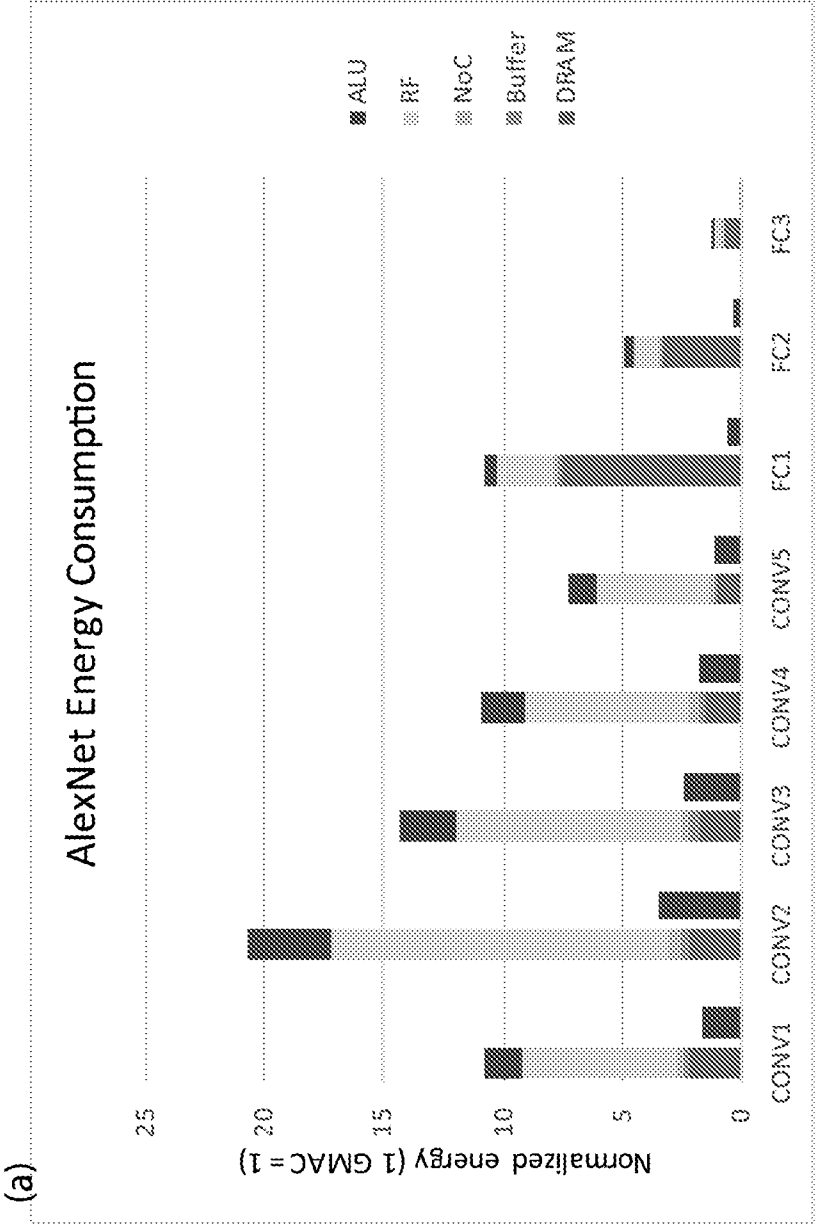


FIG. 11A

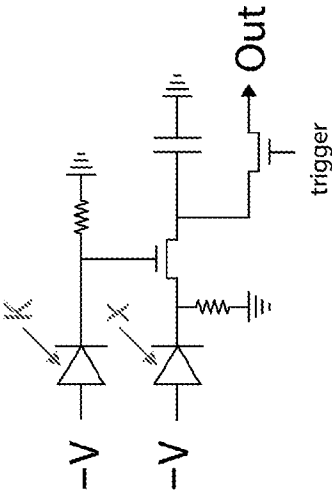


FIG. 11B

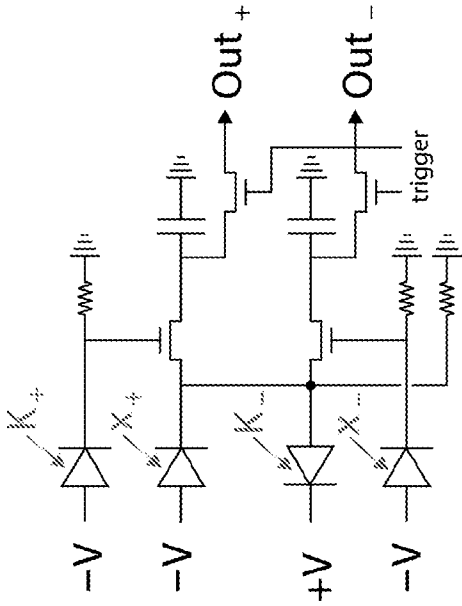


FIG. 11C

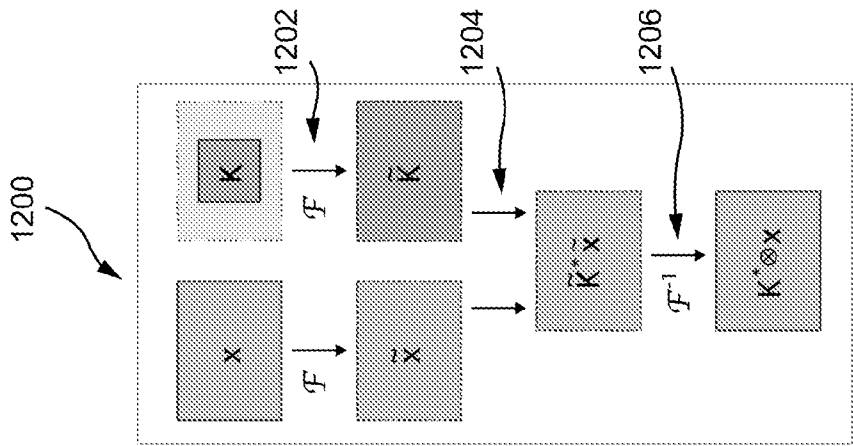


FIG. 12A

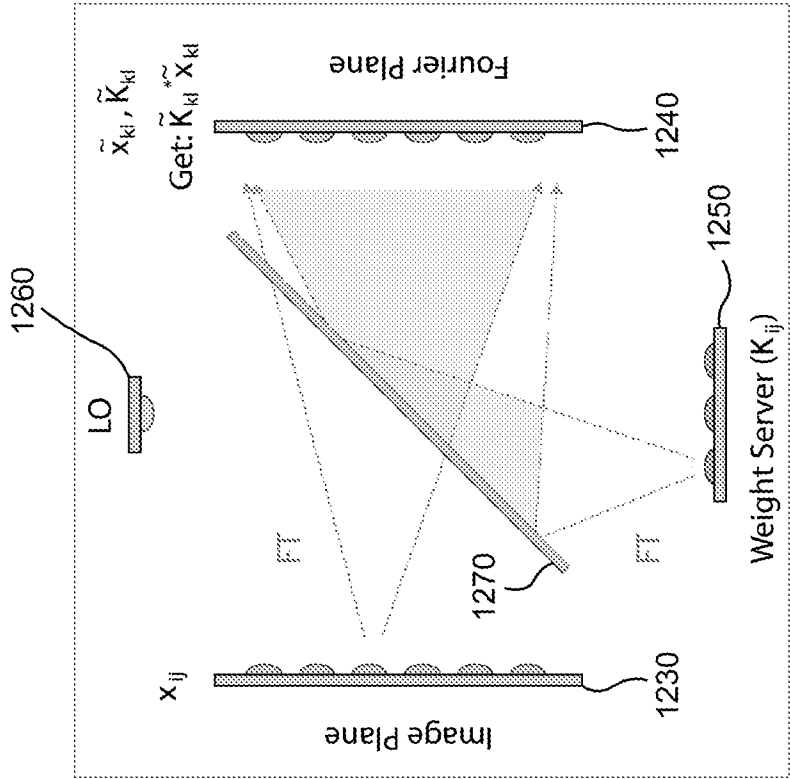


FIG. 12B

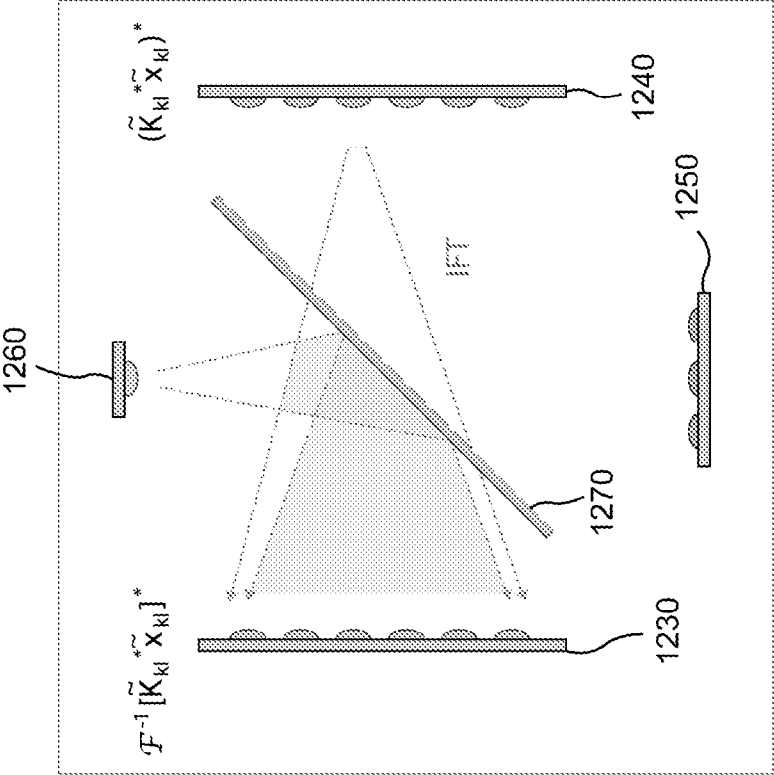


FIG. 12C

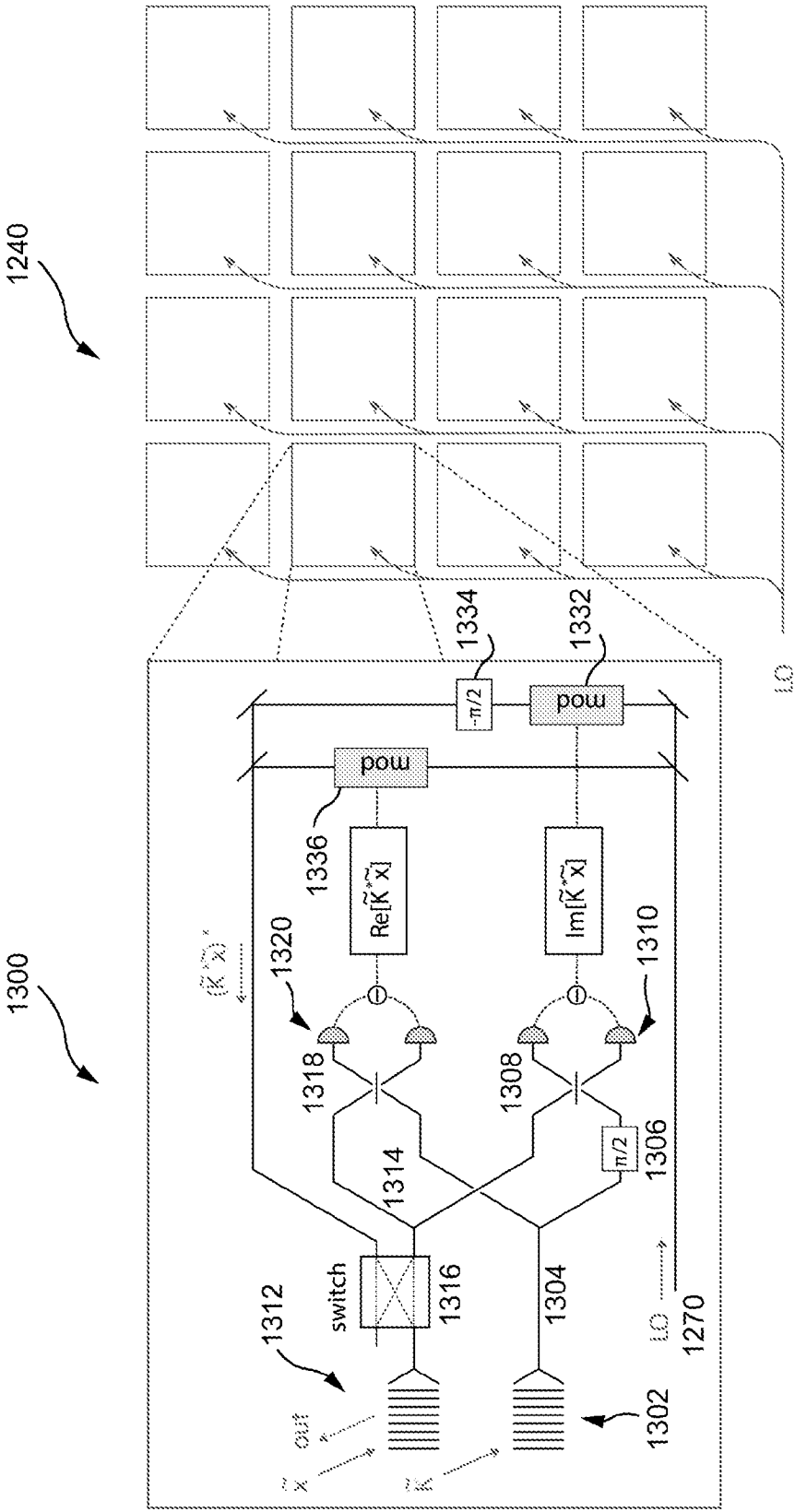


FIG. 13

FIG. 14A

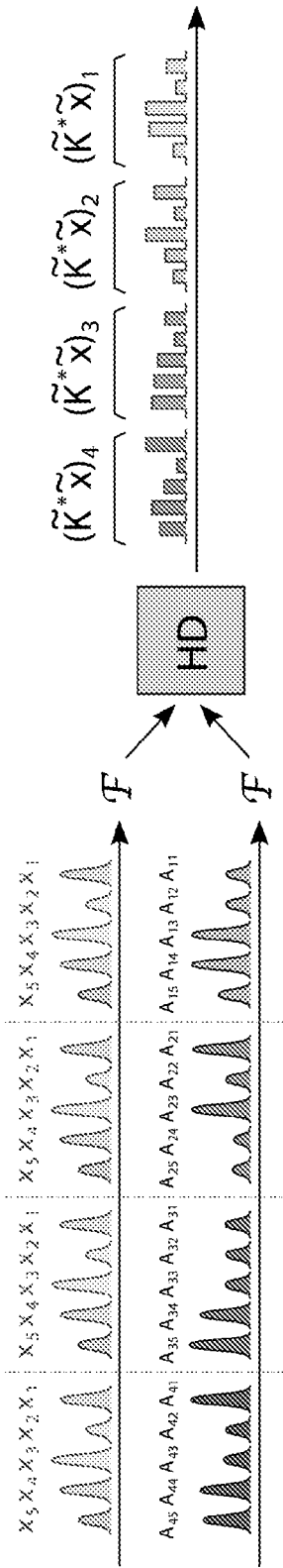
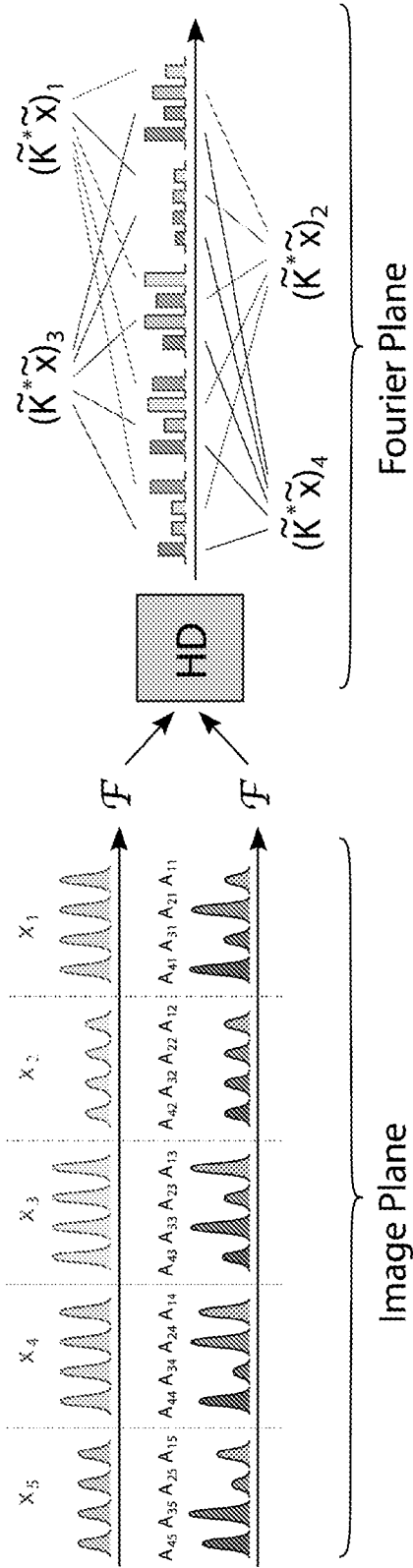


FIG. 14B

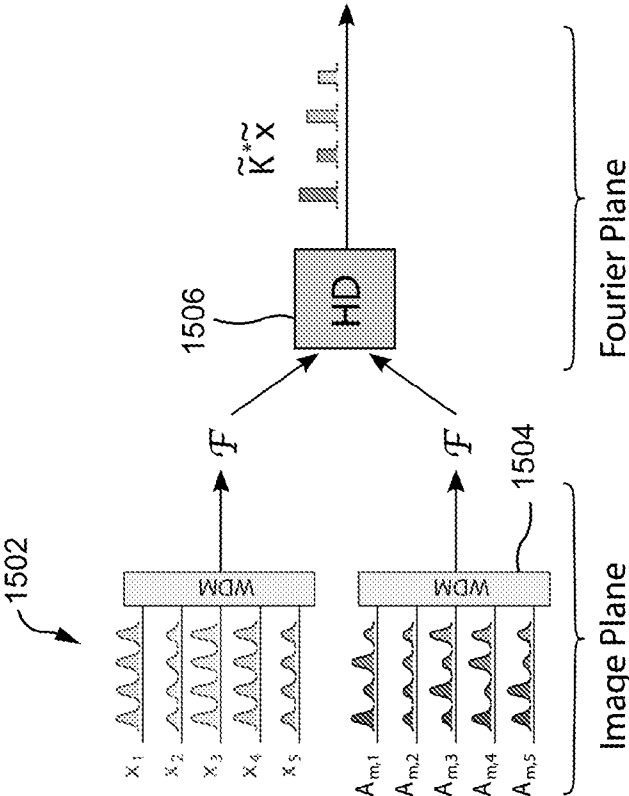


FIG. 15A

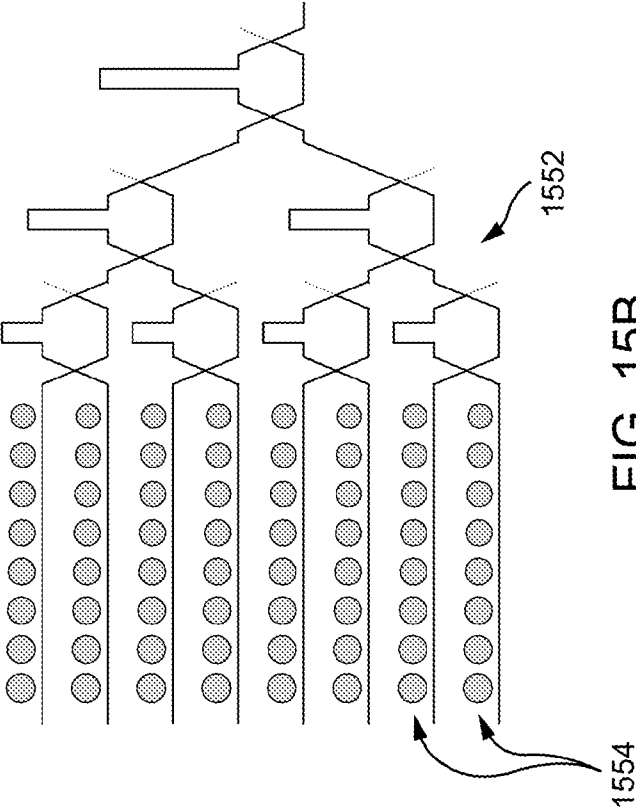


FIG. 15B

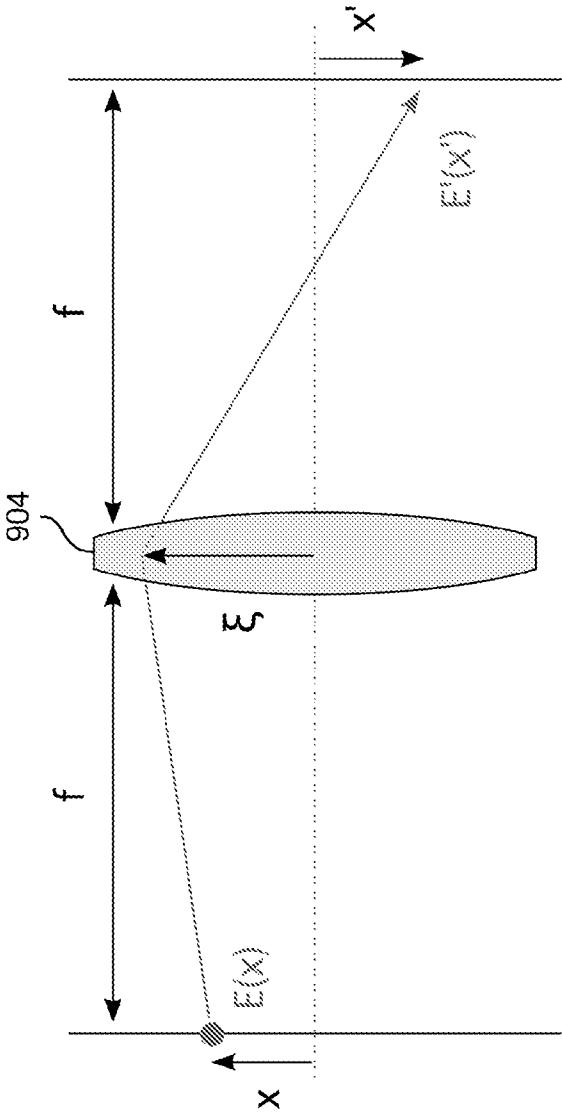


FIG. 16

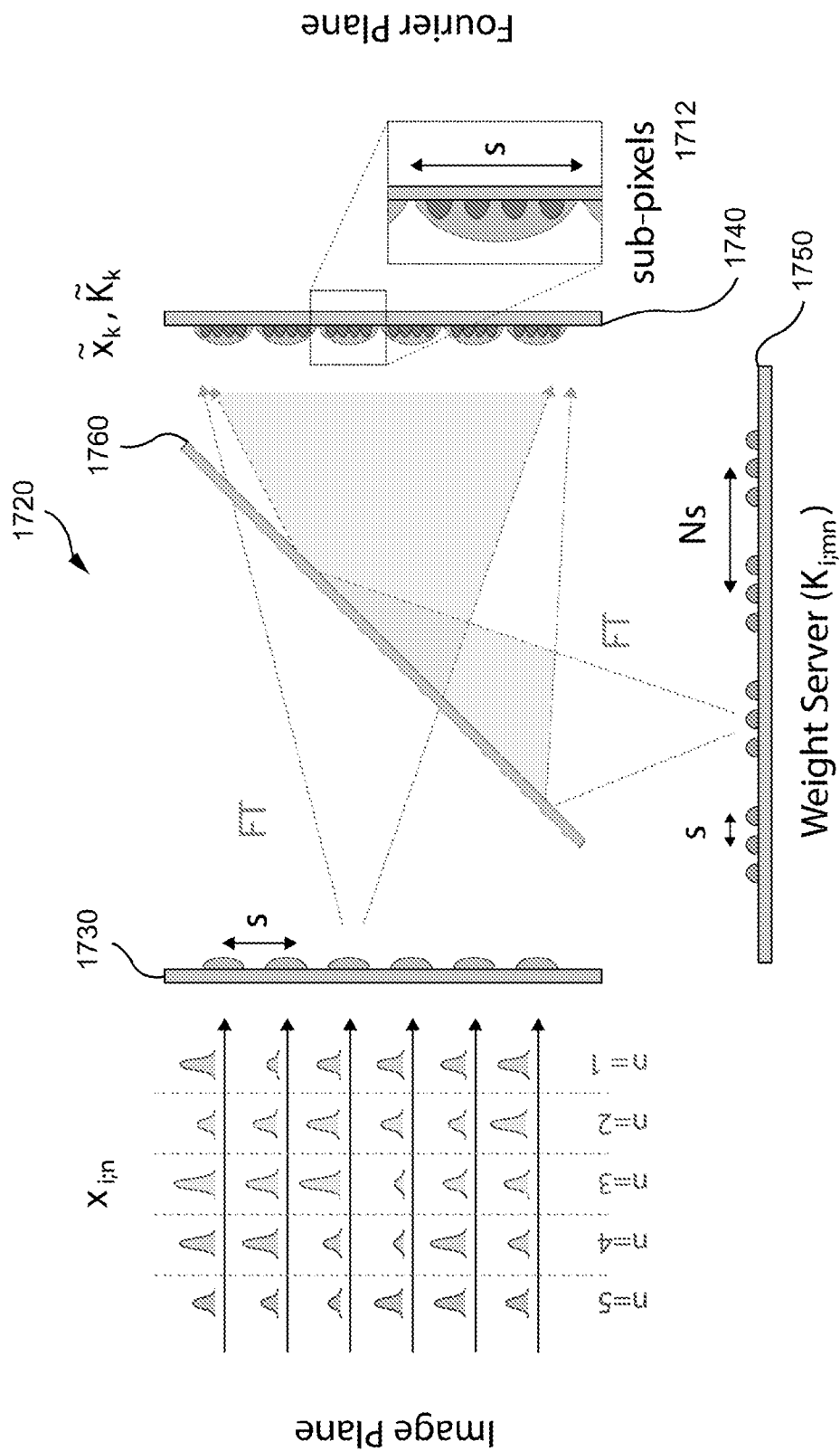


FIG. 17

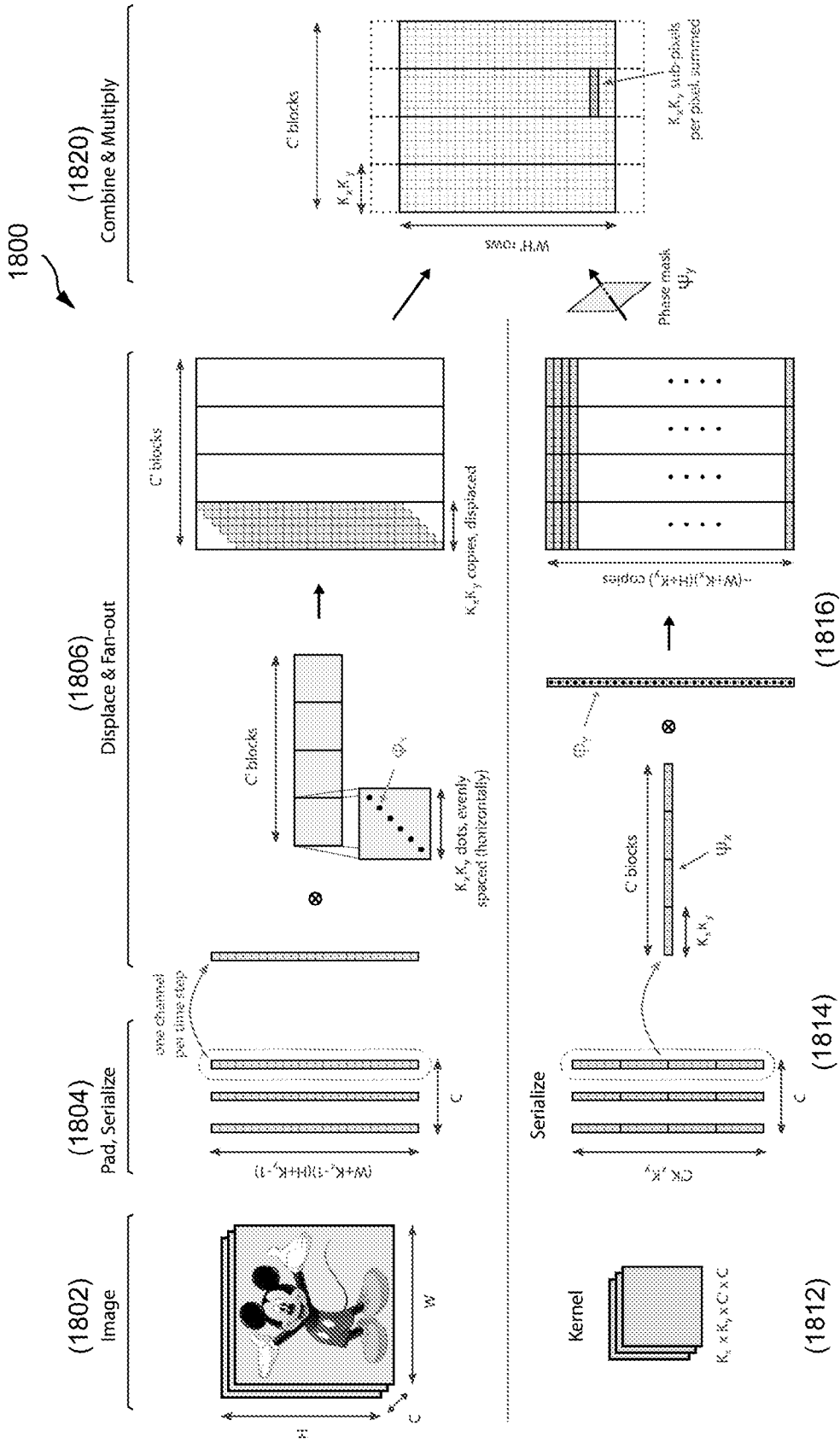


FIG. 18

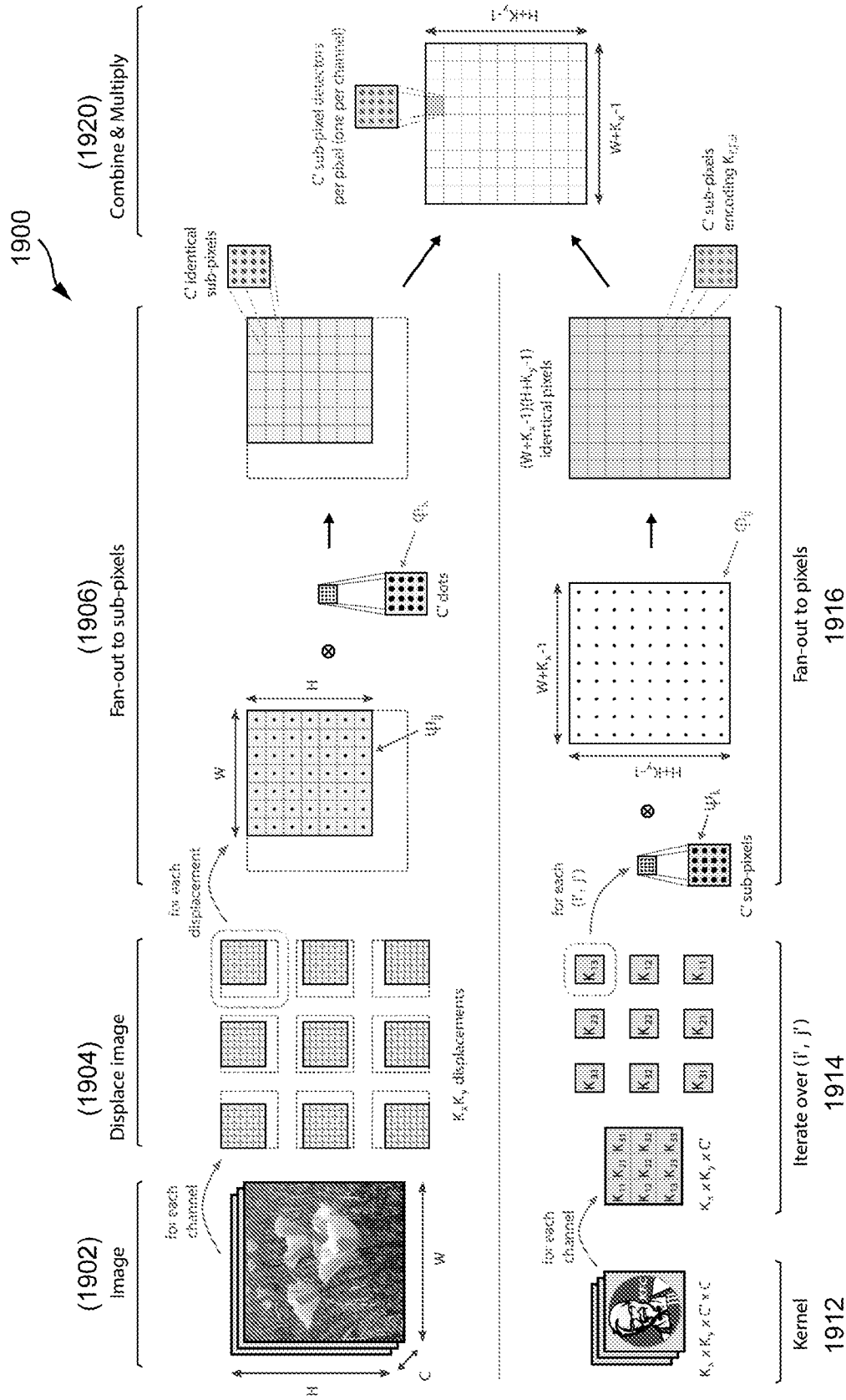


FIG. 19

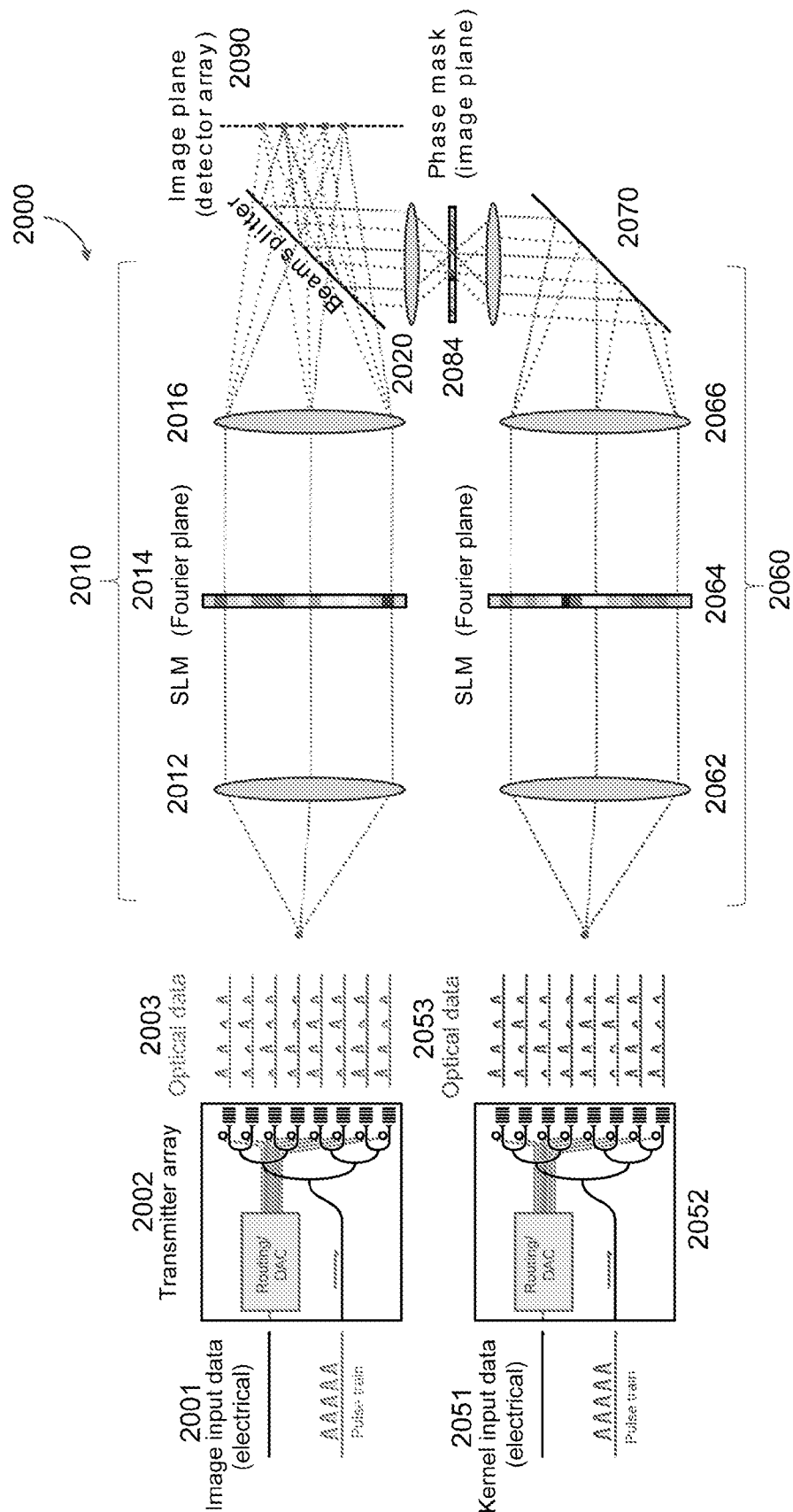


FIG. 20

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 19/60935

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
see extra sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-20

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- ☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- ☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 19/60935

A. CLASSIFICATION OF SUBJECT MATTER
 IPC - G06N 3/04; G06N 3/067 (2020.01)
 CPC - G06E 3/008; G06N 3/04; G06N 3/0675

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
 See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
 See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
 See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X ----- P, A	US 2019/0244090 A1 (ENGLUND) 08 August 2019 (08.08.2019) Fig 1A-1E, 2-5, 6A, 6B, abstract, para [0004]-[0066]	1, 5-12, 14-16 ----- 2-4, 13, 17-20
P, A	US 2019/0019100 A1 (ROQUES-CARMES et al.) 17 January 2019 (17.01.2019) Fig 1-3, 4A, 4B, 5A, 5B, 6A, 6B, 7, para [0003]-[0058]	1-20
A	US 4,959,532 A (OWECHKO) 25 September 1990 (25.09.1990) Fig 2, abstract, col 5, ln 53-col 6, ln 40	1-20
A	US 2005/0018295 A1 (MENDLOVIC et al.) 27 January 2005 (27.01.2005) Fig 2, 4, 7, abstract, para [0037], [0039], [0040], [0043]-[0045], [0089], [0090]	1-20
A	US 2018/0262291 A1 (GOVERNMENT OF THE UNITED STATES OF AMERICA, AS REPRESENTED BY THE SECRETARY OF THE NAVY) Fig 1, 2, abstract, para [0008], [0020], [0021], [0023]	1-20
A	JARAJREH, M.A., et al. Artificial Neural Network Nonlinear Equalizer for Coherent Optical OFDM. IEEE Photonics Technology Letters [online]. Vol. 27, No. 4, 15 February 2015 (15.02.2015) [retrieved on 2020-02-24]. Retrieved from the Internet: <URL: https://ieeexplore.ieee.org/abstract/document/6975096>.	1-20

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance
 "D" document cited by the applicant in the international application
 "E" earlier application or patent but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

24 February 2019

Date of mailing of the international search report

10 MAR 2020

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
 P.O. Box 1450, Alexandria, Virginia 22313-1450
 Facsimile No. 571-273-8300

Authorized officer

Lee Young

Telephone No. PCT Helpdesk: 571-272-4300

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/US 19/60935

Continuation of Box No III Observations where unity of invention is lacking

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be searched, the appropriate additional search fees must be paid.

Group I: Claims 1-20 drawn to an optical neural network, method, or convolutional layer including homodyne processing of input signals and weight signals.

Group II: Claim 21 drawn to a method of fanning out data in a digital optical neural network including breaking images into patches.

The inventions listed as Groups I and II do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

Special Technical Features:

Group I includes special technical features including a two-dimensional array of homodyne receivers, in optical communication with the first optical fan-out element and the second optical fan-out element, to produce electronic signals representing interference of each of the N copies of the M optical input signals with a corresponding copy of the M copies of the N optical weight signals (claims 1-11), transducing, with an array of homodyne receivers, interference of each of the N copies of the M optical input signals with a corresponding copy of the M copies of the N optical weight signals into electronic signals (claim 12-16), or a Fourier-plane transceiver array, in a Fourier plane of the image-plane transceiver array and in optical communication with the beam splitter, to detect a homodyne product of a spatial Fourier transform of the array of input signals and a spatial Fourier transform of the array of weight signals (claims 17-20), not included in the other group.

Group II includes the special technical features of, for each image in a set of images, the set of images representing an input to a layer in the digital optical neural network, breaking the image into a set of patches, each patch in the set of patches comprising $K_x \times K_y$ pixels; and for each patch in the set of patches, spatially convolving the patch with a box function; spatially convolving a corresponding kernel with a size of $K_x \times K_y$ with a lattice function having a horizontal lattice spacing of K_x and a vertical lattice spacing of K_y ; and imaging the patch and the corresponding kernel onto a detector array, not included in the other group.

Common Technical Features:

The only technical features shared by Groups I-II that would otherwise unify the groups, are an optical neural network including fanning out and convolution. However, these shared technical features do not represent a contribution over prior art, because the shared technical features are disclosed by US 2018/0262291 A1 to Government of the United States of America, as represented by the Secretary of the Navy (hereinafter 'US Sec Navy').

US Sec Navy discloses an optical neural network (abstract) and suggests including optical fan out (para [0008], "...Improvements can be made...utilizing beam copy through standard fanout holograms.") and implementing the optical neural network as a convolutional neural network (Fig 2, para [0023], "...neural network 150 includes...Convolutional Neural Network..").

As the shared technical features were known in the art at the time of the invention, they cannot be considered special technical features that would otherwise unify the groups.

Therefore, Groups I-II lack unity under PCT Rule 13.