UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CROWDSTRIKE, INC.,
Petitioner

v.

SKYSONG INNOVATIONS, LLC,
Patent Owner

Inter Partes Review Case No. IPR2025-01399
U.S. Patent No. 11,775,831

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 11,775,831

TABLE OF CONTENTS

I. INTRODUCTION 1

II. THE '831 PATENT 1

A. THE '831 PATENT'S ALLEGED INVENTION 1
B. THE '831 PATENT'S PROSECUTION 4
C. LEVEL OF SKILL OF A PERSON HAVING ORDINARY SKILL IN THE ART 4

III. REQUIREMENTS UNDER 37 C.F.R. § 42.104 5

A. GROUNDS FOR STANDING UNDER 37 C.F.R. § 42.104(A) 5
B. IDENTIFICATION OF CHALLENGE UNDER 37 C.F.R. § 42.104(B) 5
C. CLAIM CONSTRUCTION UNDER 37 C.F.R. § 42.104(B)(3) 6

IV. PRIOR ART SUMMARIES 6

A. UJIIE 6
B. MOONS 9
C. KAUL 10
D. ISHII 11
E. RAVINDRAN 12

V. GROUND 1: CLAIMS 1, 10, AND 11 ARE OBVIOUS OVER UJIIE AND MOONS 12

A. CLAIM 1 12
1. Claim 1(Pre): "One or more non-transitory computer-readable storage media storing instructions which, when executed by at least one processor, cause the at least one processor to perform operations comprising:" 12
2. Claim 1(a): "in one or more layers of a convolutional neural network (CNN), performing a first iteration that includes computing a value based on a first set of most significant bits (MSBs) for each of a plurality of data sets;" 14
a) Motivation to Combine Ujiie and Moons 18
3. Claim 1(b): "examining a first set of values computed for the plurality of data sets in the first iteration to determine whether a maximum value is present among the first set of values;" 24

4. Claim 1(c): "responsive to identifying the maximum value, performing a full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value; and" *26*
5. Claim 1(d): "propagating the full precision computation of the value to a subsequent layer of the CNN." *27*
B. CLAIM 10 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 1, WHEREIN THE CNN IS EMPLOYED TO ANALYZE AN IMAGE. 29
C. CLAIM 11 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 1, WHEREIN: 30
1. Claim 11(a): "the first iteration computes a value that approximates the full precision computation of the value; and" *30*
2. Claim 11(b): "the full precision computation is performed on the data set that includes less data than the plurality of data sets." *31*

VI. GROUNDS 2 AND 3: CLAIMS 2-6, 8-9 ARE OBVIOUS OVER UJIIE, MOONS, AND KAUL, OR ALTERNATIVELY, CLAIMS 2-6, 8-9 ARE OBVIOUS OVER UJIIE, MOONS, KAUL, AND ISHII 32

A. CLAIM 2 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 1, FURTHER COMPRISING: RESPONSIVE TO DETERMINING THAT THE FIRST SET OF VALUES ARE THE SAME, PERFORMING, BY THE AT LEAST ONE PROCESSOR, A SECOND ITERATION THAT INCLUDES COMPUTING THE VALUE BASED ON A SECOND SET OF MSBS FOR EACH OF THE PLURALITY OF DATA SETS, THE SECOND SET OF MSBS BEING LARGER THAN THE FIRST SET OF MSBS. 32
1. The '831 Inventors Cited to Kaul's Adaptive Precision Process in a Related Publication, Conceding That Iteratively Increasing Bit-Widths in Neural Networks was not new *32*
2. Ujiie-Moons in View of Kaul *33*
a) Motivation to Combine *Ujiie*, *Moons*, and *Kaul* 36
3. Ujiie-Moons-Kaul in View of Ishii *39*
a) Motivation to Combine *Ujiie*, *Moons*, *Kaul*, and *Ishii* 42
B. CLAIM 3 44

1. Claim 3(a): "The one or more non-transitory computer-readable storage media of claim 2, further storing instructions, which, when executed by the at least one processor, cause the at least one processor to perform operations comprising: examining a second set of values computed for the plurality of data sets in the second iteration to determine whether the maximum value is present among the second set of values; and" 44

2. Claim 3(b): "responsive to identifying the maximum value among the second set of values, performing, by the at least one processor, the full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value in the second iteration." 45

C. CLAIM 4 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 2, WHEREIN THE COMPUTING IN EACH OF THE FIRST ITERATION AND THE SECOND ITERATION EMPLOYS A CONVOLUTION AND A POOLING. 45

D. CLAIM 5 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 4, WHEREIN THE CONVOLUTION IS A NXN CONVOLUTION, WHERE N IS ANY INTEGER. 46

E. CLAIM 6 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 4, WHEREIN THE POOLING IS A NXN POOLING, WHERE N IS ANY INTEGER. 47

F. CLAIM 8 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 2, WHEREIN AT LEAST ONE OF THE FIRST ITERATION AND THE SECOND ITERATION IS PERFORMED WITH A PRECISION LESS THAN THAT OF THE FULL PRECISION COMPUTATION. 48

G. CLAIM 9 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 8, WHEREIN THE PRECISION IS 8-BIT PRECISION. 49

VII. GROUNDS 4 AND 5: CLAIM 7 IS OBVIOUS OVER UJIIE, MOONS, KAUL, AND RAVINDRAN OR ALTERNATIVELY, CLAIM 7 IS OBVIOUS OVER UJIIE, MOONS, KAUL, ISHII, AND RAVINDRAN 52

A. CLAIM 7 – THE ONE OR MORE NON-TRANSITORY COMPUTER-READABLE STORAGE MEDIA OF CLAIM 4, WHEREIN THE

CONVOLUTION IS A 3X3 CONVOLUTION, AND THE POOLING IS A 2X2 POOLING.....52
a) Motivation to Combine Ujiie, Moons, Ishii, Kaul, and Ravindran 54
VIII. CONCLUSION.....55
IX. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1) 56
A. REAL PARTIES-IN-INTEREST 56
B. RELATED MATTERS 56
C. LEAD AND BACK-UP COUNSEL 56
D. 37 C.F.R. § 42.8(B)(4) – SERVICE INFORMATION.....57

TABLE OF AUTHORITIES

Cases

Phillips v. AWH Corp., 415 F.3d 1303 (Fed. Cir. 2005).....6

Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc., Case No. 7:25-cv-00040-DC-DTG 56

Skysong Innovations, LLC v. Fortinet, Inc., Case No. 2:25-cv-00098.....56

Statutes

35 U.S.C. §102(a)(1) 7, 9, 10, 11, 12

35 U.S.C. § 102(a)(2) 11,12

Regulations

37 C.F.R. § 42.100(b).....6

37 C.F.R. § 42.104 5

37 C.F.R. § 42.104(b).....5

37 C.F.R. § 42.104(b)(3) 6

37 C.F.R. § 42.8(a)(1) 56

37 C.F.R. § 42.8(b)(1) 56

37 C.F.R. § 42.8(b)(3) 56

37 C.F.R. § 42.8(b)(4) 57

37 C.F.R. §§ 42.6(e) 64

37 C.F.R. §42.24 63

37 C.F.R. §42.8 63

I. INTRODUCTION

CrowdStrike, Inc. ("Petitioner") respectfully requests inter partes review ("IPR") of claims 1-11 ("Challenged Claims") of U.S. Patent No. 11,775,831 ("'831 Patent").

II. THE '831 PATENT

A. The '831 Patent's Alleged Invention

The '831 Patent describes techniques for "reducing the amount of total computation in convolutional neural networks (CNNs) without affecting the output result or classification accuracy." *'831 Patent* (Ex. 1001), Abstract, 1:33-36. The claimed approach first performs a low precision convolution on a number of datasets, determines which dataset exhibits a maximum value, and performs a "full-precision convolution on the entry that is the maximum value." *Id.*, 3:32-37, claim 1. This approximate convolution first, full precision convolution second, technique is depicted below in Figure 4:

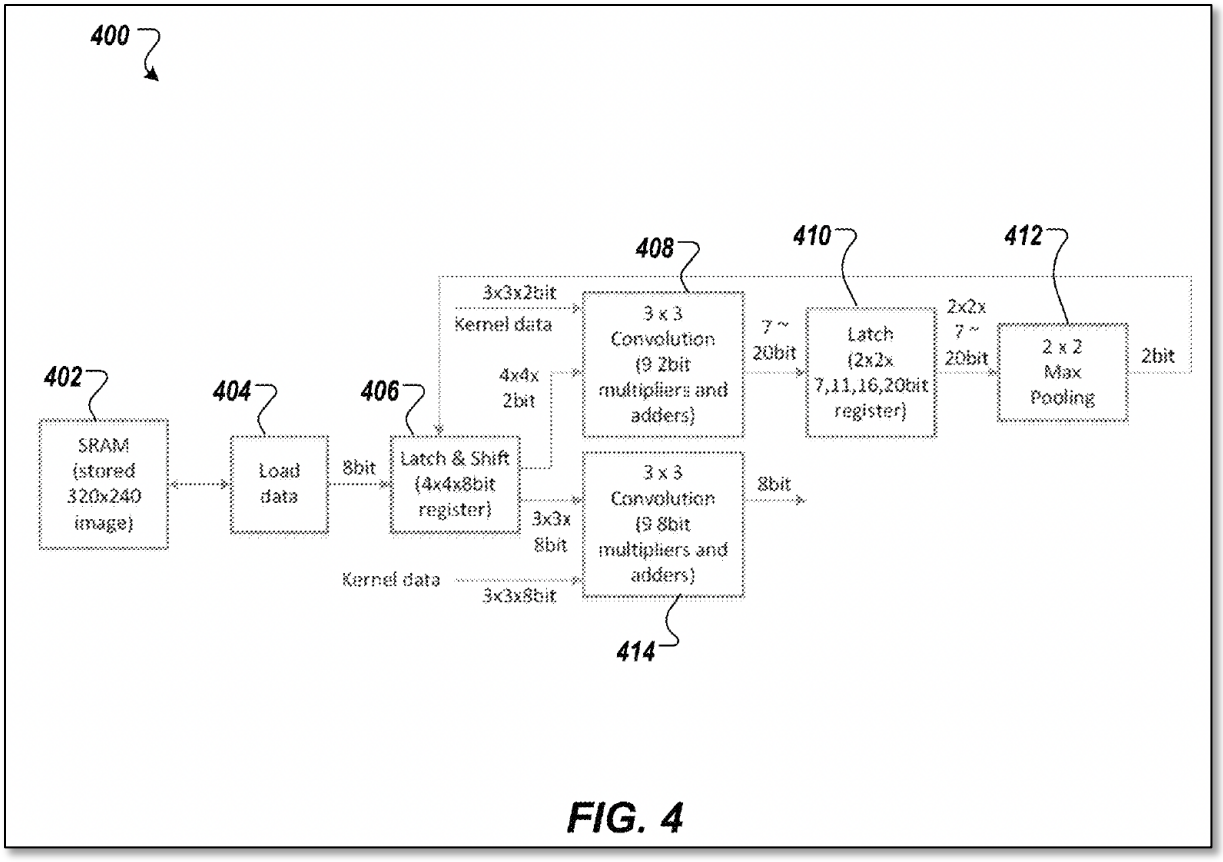


FIG. 4

Id., Fig. 4.

As shown in Figure 4, an initial low precision convolution is performed to identify datasets from the image data that exhibit maximum values. "Using the two most significant bits of image data and convolution kernel, approximate multiply-and-add operations are performed (408)." *Id.*, 7:7-23. "The intermediate approximate results are stored into registers (410), and 2x2 max-pooling is performed on the approximate 2-bit multiply-and-add results (412)." *Id.* "If a maximum is found, the 3x3 convolution may be performed and full precision convolution may be performed on the maximum-exhibiting data set (414)." *Id.* Otherwise, "[i]f a maximum is not found," the low precision convolution process is

repeated at a "higher precision[,]" using more bits than the first low precision convolution. This process repeats iteratively "until a maximum is found." *Id*.

According to the '831 Patent, this contrasts the "conventional convolution and pooling scheme" shown in flow diagram 200 below:

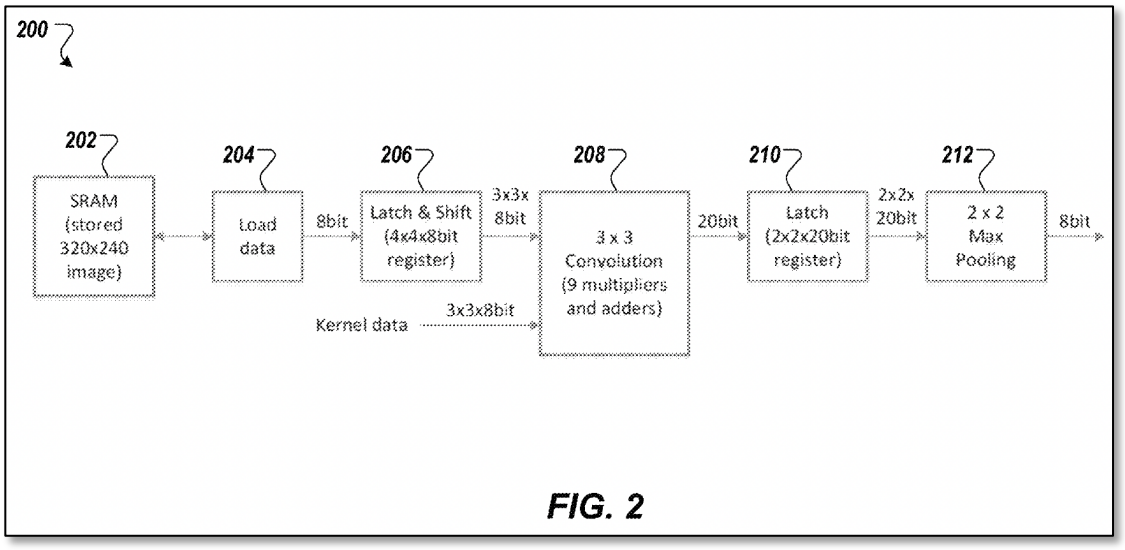


FIG. 2

Id., Fig. 2. This exemplary conventional CNN "employs 3x3 convolution, 2x2 pooling, and 8-bit precision." *Id*., 5:65-6:1. Here, "an image (e.g., 320x240) may be stored (202) in SRAM[,]" and loaded 204 into registers 206 for processing. *Id*., 6:12:25. "Convolution (208) may be performed on the data in the registers, e.g., 3x3 convolution based on 3x3 kernel data (e.g., weighting)." *Id*. "The convolution results may be stored in 2x2 bit registers (210), and 2x2 max pooling (212) may be performed to determine the maximum value for the particular layer of analysis before proceeding to the next layer." *Id*.

These conventional methods performed full precision convolutions on every dataset, leading to increased computational and energy expenditure. *Id.*, 3:10-24. The goal of the claimed invention is to "perform the more fine-grain, full precision, (e.g., more accurate) computations on the narrowed-down set of data that exhibited the maximum" to reduce the number of redundant full-precision computations and energy used in the claimed CNN processes. *Id*., 4:1-4, 9:61-64.

B. The '831 Patent's Prosecution

The '831 Patent claims its earliest priority to provisional application 62/399,753, filed on September 26, 2016. For purposes of this Petition only, Petitioner applies September 26, 2016, as the priority date for the Challenged Claims.

The application received a first action Notice of Allowance after an Examiner's Amendment to add a missing comma in the pending claims. *'831 File History*, (Ex. 1002), 136-150.

C. Level of Skill of a Person Having Ordinary Skill in the Art

A POSITA at the time of the '831 Patent would have had a bachelor's degree in electrical and/or computer engineering, computer science, or the equivalent, and 2-3 years of work/research experience in convolutional neural networks, artificial

neural networks, deep learning, or deep neural networks. *Dec.*, ¶¶41-43.[1] Additional education (e.g., MS degree) might substitute for less experience and vice versa.

III. REQUIREMENTS UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies the '831 Patent is eligible for IPR and Petitioner is not barred or estopped from requesting IPR challenging the claims of the '831 Patent. Petitioner is not the owner of the '831 Patent, has not filed a civil action challenging the validity of any claim of the '831 Patent, and this Petition is filed less than one year after Petitioner was served with a complaint alleging infringement of the '831 Patent.

B. Identification of Challenge Under 37 C.F.R. § 42.104(b)

Petitioner requests the Challenged Claims be found unpatentable on the following grounds.

Proposed Grounds of Unpatentability	Exhibit(s)
Ground 1: Claims 1, 10, and 11 are **obvious** under AIA 35 U.S.C. § 103 over *Ujiie* and *Moons*	Ex. 1004, Ex. 1005
Ground 2: Claims 2-6, 8-9 are **obvious** under AIA 35 U.S.C. § 103 over *Ujiie*, *Moons*, and *Kaul*	Ex. 1004, Ex. 1005, Ex. 1006
Ground 3: Claims 2-6, 8-9 are **obvious** under AIA 35 U.S.C. § 103 over *Ujiie*, *Moons*, *Kaul*, and *Ishii*	Ex. 1004, Ex. 1005,

[1] All citations to "Dec." are to paragraph numbers in Ex. 1003, Declaration of Dr. Eli Saber.

	Ex. 1006, Ex. 1007
Ground 4: Claim 7 is **obvious** under AIA 35 U.S.C. § 103 over *Ujiie*, *Moons*, *Kaul*, and *Ravindran*	Ex. 1004, Ex. 1005, Ex. 1006, Ex. 1008
Ground 5: Claim 7 is **obvious** under AIA 35 U.S.C. § 103 over *Ujiie*, *Moons*, *Kaul*, *Ishii*, and *Ravindran*	Ex. 1004, Ex. 1005, Ex. 1006, Ex. 1007, Ex. 1008

C. Claim Construction Under 37 C.F.R. § 42.104(b)(3)

In this proceeding, claims are interpreted under the *Phillips* standard. 37 C.F.R. § 42.100(b); 83 Fed. Reg. 197 (Oct. 11, 2018); *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (*en banc*). Petitioner applies the ordinary and customary meaning of the claim terms as understood by a POSITA.[2]

IV. PRIOR ART SUMMARIES

A. Ujiie

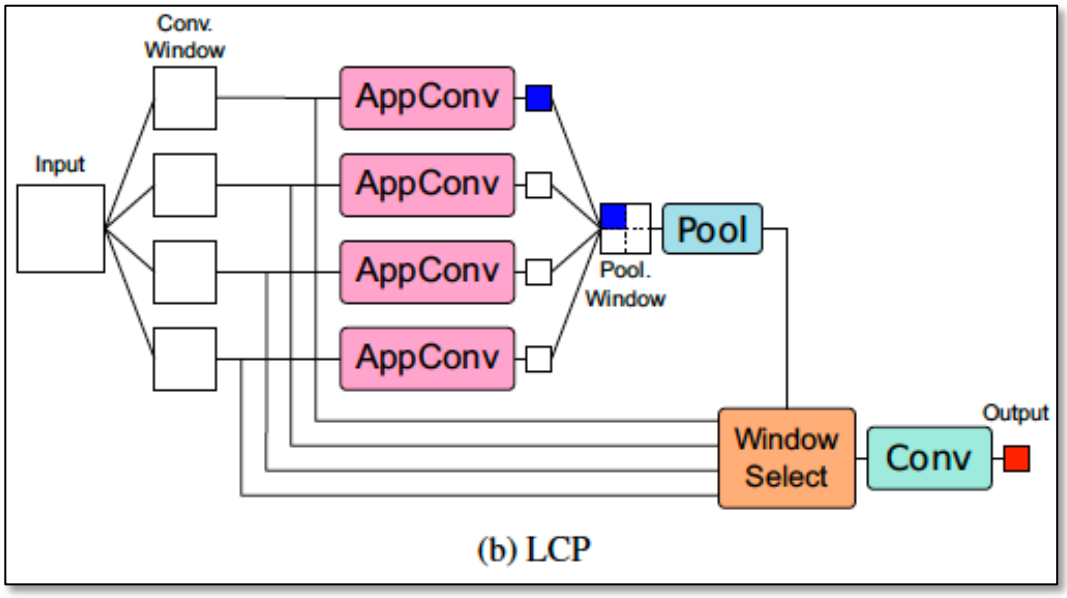
"Approximated Prediction Strategy for Reducing Power Consumption of Convolutional Neural Network Processor" by Ujiie et al. ("*Ujiie*") was published in the 29th IEEE Conference on Computer Vision and Pattern Recognition Workshops,

[2] Petitioner's challenge is not a waiver of any argument in litigation that any claim term in the '831 Patent is indefinite or invalid. Petitioner does not waive its right to seek to construe claim terms in a different proceeding.

and was publicly available at least by August 5, 2016. Ex. 1004; *see also Munford Dec.* (Ex. 1009), ¶¶11-26; *Dec.*, ¶¶61-65. Namely, *Ujiie* was publicly available at least as of August 5, 2016, on Computer Vision Foundation's (CVF) open access webpage. *Munford Dec.*, ¶¶16-24. Ms. June Munford located CVF's open access webpage for *Ujiie* on WebArchive and analyzed the archived date—August 5, 2016. *Id*. As of August 2016, CVF's open access webpage allowed interested users to search for references using a variety of search parameters. *Id*. This same search functionality is available today. *Id*. *Ujiie* is available in the CVF's open access webpage; was available since at least August 5, 2016; and simple search queries including keywords from the title and abstract led to locating *Ujiie*. *Id*. This methodology would have led an interested party to locate *Ujiie*. *Dec.*, ¶¶61-65. Moreover, *Ujiie* was made available at the 2016 Conference on Computer Vision and Pattern Recognition, at least by the end of the conference on July 1, 2016. *Munford Dec.*, ¶¶12-13; *MacPherson Dec.*, (Ex. 1010), ¶¶6-9, 13. *Ujiie* is prior art to the '831 Patent under at least 35 U.S.C. § 102(a)(1).

Ujiie is generally directed to an "approach[] for realizing energy-efficient CNN[s]" by reducing "the redundancy in most of the recognition algorithms[.]" *Ujiie*, 1. To decrease redundancy, *Ujiie* teaches "approximated convolution units (AppConv) to predict the feature selection in the pooling layer, and [then] a single

regular (and exact) convolution unit (Conv) to perform convolution only on the predicted window." *Id.*, 4. This process is illustrated in Figure 2b below:



Id., Fig. 2b.

Ujiie's process, termed "LazyConvPool (LCP)," is "an approximate computational method that combines convolution and max pooling for reducing power consumption of CNN processors." *Id.*, 3. "The computation flow of LCP is summarized as follows: Step 1 Compute AppConv using the lightweight and approximate computation. Step 2 Operate max pooling to the results of AppConv to specify the convolution index of which the convolution result is propagated. Step 3 Compute Conv only to the input corresponding to the specified index in Step 2." *Id.*, 4. Using this process, *Ujiie's* system "only executes an exact convolution (Conv) against the specified window predicted by the approximate convolution (AppConv). *Id.*, 3. *Ujiie's* approximation process "reduce[s] the number of multiplications in

CNN[,]" thereby "reduc[ing] the energy consumption for image recognition by 11.4%–14.1% while keeping the recognition accuracy." *Id.*, 6.

B. Moons

"Energy-Efficient ConvNets Through Approximate Computing" by Bert Moons et al. ("*Moons*") published February 17, 2016, according to its MARC record. *Munford Dec.*, ¶¶27-34; *MacPherson Dec.*, ¶¶6-7, 14 (establishing public availability "no later than" March 10, 2016, and public availability via IEEE Xplore by May 26, 2016). *Moons* is prior art to the '831 Patent under at least 35 U.S.C. § 102(a)(1).

Moons "proposes methods based on approximate computing to reduce energy consumption in state-of-the-art ConvNet accelerators." *Moons*, 1. *Moons* "quantize[s] existing networks and drastically reduce[s] the number of bits that encode the weights and inputs of each layer" while maintaining accuracy. *Id*. This both "can be used to reduce energy consumption in many neural network applications and even dynamically scale energy consumption depending on the current application." *Id*. Specifically, *Moons* employs "precision scaling, in which the number of bits used in computations is varied at run-time[.]" *Id*., 2. "By scaling precision (i.e., dynamically scaling the number of bits that encode the network's weights and inputs)," power consumption is reduced. *Id*.; *see also id.*, 5, Fig. 1. For example, *Moons* teaches quantizing weights and inputs to "4-10 bits instead of the

previously used 16- or 32-bit formats" so long as the "MSB-bit" is "placed at the MSB position." *Id*., 5, 6.

C. Kaul

"A 21.5M-Query-Vectors/s 3.37nJ/Vector Reconfigurable k-Nearest-Neighbor Accelerator with Adaptive Precision in 14nm Tri-Gate CMOS" by Kaul et al. ("*Kaul*") was published in 2016 IEEE International Solid-State Circuits Conference (ISSCC) and published on February 17, 2016, according to its MARC record. *Munford Dec.*, ¶¶35-42; *MacPherson Dec.*, ¶¶6-7, 15 (establishing public availability "no later than" February 4, 2016, and public availability via IEEE Xplore by February 25, 2016). Therefore, *Kaul* is prior art to the '831 Patent under 35 U.S.C. § 102(a)(1).

Kaul is generally directed to "[a]daptive precision" for "[e]nergy-efficient k-nearest-neighbor (kNN) computations [which] are key building blocks for computer vision, classification, and machine-learning workloads[.]" *Kaul*, 1. Using adaptive precision, *Kaul* "improves energy efficiency by eliminating a majority of vectors without costly full-precision computation, with as-needed precision refinement[.]" *Id.* To this aim, *Kaul's* system uses "MSB-to-LSB-based selective distance refinement" which "start[s] with low-accuracy from the MSB" and then performs "[i]terative refinement of accumulated distance…as needed with successive sort operations on 2b windows from MSB to LSB." *Id.* This method allows for "[e]arly

removal of vectors while moving from MSB to LSB for nearest-neighbor search result[ing] in 47% compute reduction[.]" *Id.*

D. Ishii

U.S. Patent Application Publication No. 2016/0259995 to Ishii et al. ("*Ishii*") was filed on February 22, 2016, and published on September 8, 2016, qualifying as prior art to the '831 Patent under 35 U.S.C. §§ 102(a)(1) and 102(a)(2). *Ishii*, (Ex. 1007).

Ishii is generally directed to "an image recognition method and the like that enables high-speed and high-precision image recognition." *Ishii*, [0006]. *Ishii's* image recognition system includes "convolution processing" and "subsampling [i.e., "Max-pooling"] processing." *Id.*, [0059], [0073]. *Ishii's* convolutional processing includes "low-resolution[,]" "medium-resolution[,]" and "high-resolution" processing, where the system "select[s] a processing result having a maximum value from among a plurality of values" from the various convolutions. *Id.*, [0102], [0119], Fig. 4. "[I]n a case where there are two or more maximum values, the output selecting unit 124 need just select any one of the two or more maximum values." *Id.*, [0130]. *Ishii* similarly teaches that in the event of max-pooling, if two or more values are the same, the system will proceed with one of those values. *Id.*, [0142], Fig. 5.

E. Ravindran

U.S. Patent Application Publication No. 2016/0259994 to Ravindran et al. ("*Ravindran*") was filed on March 4, 2015, and published on September 8, 2016, qualifying as prior art to the '831 Patent under 35 U.S.C. §§ 102(a)(1) and 102(a)(2). *Ravindran* (Ex. 1008).

Ravindran is generally directed to "[a]n image processing system…[that] builds and trains an ensemble of deep learning models, such as convolutional neural networks (CNNs), to accurately and automatically perform image processing to detect particular attributes of objects in a digital image[.]" *Id.*, [0010]. In its method, *Ravindran* discloses various sizes of convolution and max pooling kernels. *Id.*, [0047]-[0050], Fig. 6. Specifically, *Ravindran* teaches utilizing both 5x5 and 3x3 convolutional kernels, and 2x2 max pooling. *Id.*

V. GROUND 1: CLAIMS 1, 10, AND 11 ARE OBVIOUS OVER UJIIE AND MOONS

A. Claim 1

1. Claim 1(Pre): "One or more non-transitory computer-readable storage media storing instructions which, when executed by at least one processor, cause the at least one processor to perform operations comprising:"

To the extent the preamble is limiting, *Ujiie* teaches a "base architecture of [a] **CNN processor**…[that] employ[s] a parallel architecture with a **shared memory** and **multiple processing elements**. **Each processing element, connected to the**

shared memory, executes convolution, pooling, and activation." *Ujiie*, 4[3]. Specifically, *Ujiie* teaches a "processing element architecture in which LCP [LazyConvPool] is applied to the base [CNN processor] architecture." *Id.*, 4-5. This proposed architecture is depicted below in Figure 4:

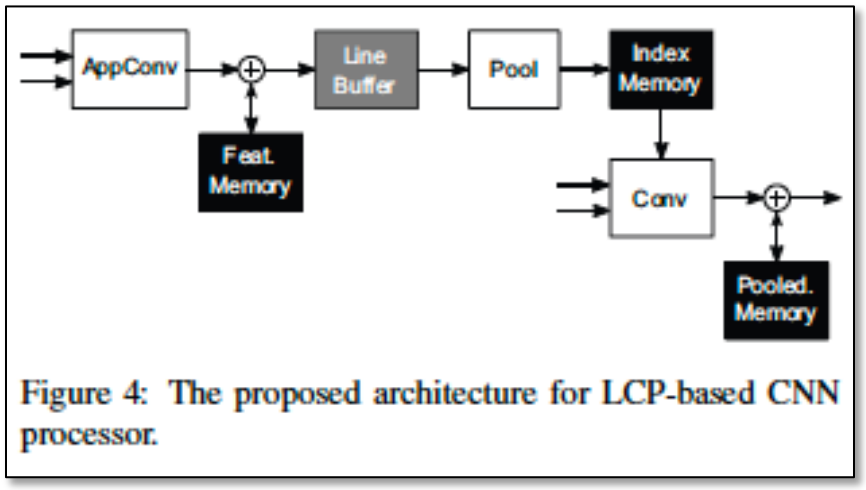


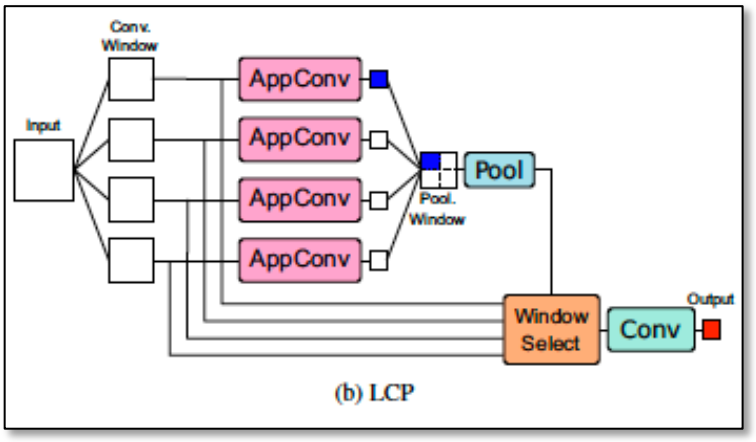
Figure 4: The proposed architecture for LCP-based CNN processor.

Id., Fig. 4. "A processing element [] computes AppConv and pooling[,]" along with the full convolution. *Id.*, 4-5. Therefore, *Ujiie* teaches a non-transitory shared memory storing instructions that when executed by its "CNN processor[,]" performs the functions of the processing elements in *Ujiie's* proposed architecture. *Dec.*, ¶¶69-70.

[3] All emphasis added unless otherwise noted.

2. Claim 1(a): "in one or more layers of a convolutional neural network (CNN), performing a first iteration that includes computing a value based on a first set of most significant bits (MSBs) for each of a plurality of data sets;"

Ujiie performs a first iteration in a convolution layer where its "LazyConvPool (LCP)" computing method computes an "approximate convolution (AppConv)" value for each dataset related to its input image. This AppConv calculation satisfies the claimed "computing a value" step (i.e., *in one or more layers of a convolutional neural network (CNN), performing a first iteration that includes computing a value*). *Ujiie*, 1-4. LCP is "a new computation method for CNN's convolution layers and pooling layers." *Id*., 1. The "LCP framework [has] two main parts: (1) **approximated convolution units (AppConv) to predict the feature selection in the pooling layer**, and (2) a single regular (and exact) convolution unit (Conv) to perform convolution only on the predicted window." *Id*., 4. Figure 2b illustrates the LCP-based convolution and pooling operations:



Id., Fig. 2b.

In more detail, "[b]uffered inputs are supplied to AppConv and Conv with weights used for convolution." *Id.*, 4. AppConv outputs "an **approximate computation unit** of reduced computation complexity." *Id*. The input image, convolution window datasets, and AppConv values are shown above in Figure 2b. *Id*., Fig. 2b; *see also id*., Tables 1 and 2 (depicting 28x28 or 32x32 image convolved by 5x5 convolution windows). To calculate these approximate convolution (AppConv) values, *Ujiie* implements an "approximate computation called Sign Connect[.]" *Id*., 4. Sign Connect quantizes the kernel weights to either +1 or -1 such that any kernel weight below zero is assigned -1 and any kernel weight above zero is assigned +1. *Id*.; *Dec.*, ¶¶72-73 (explaining that Sign Connect effectively ignores the amplitude of kernel weights, treating all negative kernel weights equally as "-1" and all positive kernel weights equally as "+1"). This process of ignoring kernel weight amplitudes greatly simplifies the processing requirements of Ujiie's AppConv, permitting "general multipliers" to be replaced "by the multiplications of the sign of weights, which can be typically realized by multiplexers." *Ujiie,* 4; *Dec.*, ¶¶72-73.

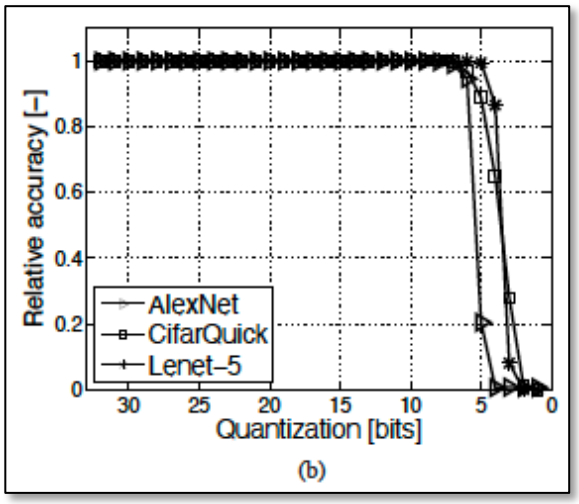
Ujiie's AppConv values are computed within a CNN layer (e.g., convolutional layer and/or convolutional and pooling layer). *Ujiie*, Tables 1 and 2 ("Layer name" "Convolution"). *Ujiie* performs a "first iteration" computing AppConv values consistent with the '831 Patent's description of an "iteration." *Id.*, Figure 2b; *Dec.*,

¶¶74-75. The '831 Patent describes "iterations" as rounds of approximate convolutions up to full precision. *'831 Patent*, 2:6-8 ("the first iteration computes a value that approximates the full precision computation of the value"), 3:58-61 ("At each iteration…a decision may be made whether the next approximate convolution iteration needs to be performed"), 4:4-6 (same), 5:50-51 ("The algorithm may perform any suitable number of iterations, up to the full precision [] convolution"), 7:65-8:2 (the process may **perform another iteration by "performing a convolution** using more bits of data"). Therefore, *Ujiie's* system performs a first iteration that includes approximate convolution values (AppConv) for each dataset. *Dec.*, ¶¶74-75.

While *Ujiie* teaches approximating convolutions using Sign Connect, *Ujiie* does not expressly discuss approximating convolutions by lowering precision "based on a first set of most significant bits[.]" Instead, as set forth above, Ujiie's Sign Connect process uses only a single bit for kernel weights and preserves the full precision values of the image data. *Moons*, however, teaches approximating convolutions via "precision scaling," which lowers the bit count of both the kernel weights and the original data inputs.

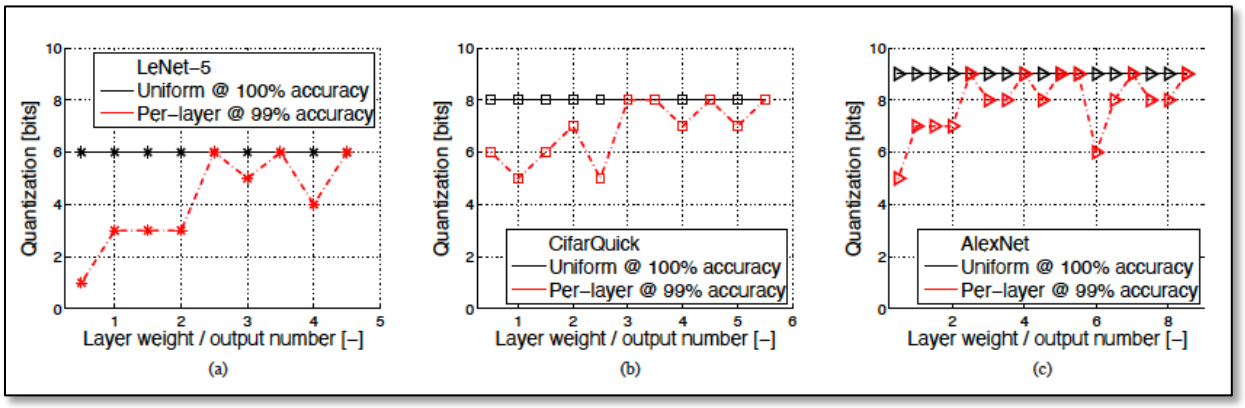
To appropriately address the "increase in energy consumption" in CNNs, *Moons* utilizes "precision scaling," which encompasses "quantiz[ing] existing networks and drastically **reduc[ing] the number of bits that encode the weights**

and inputs of each layer." *Moons*, 1. "Precision scaling is an approximate computing technique allowing major energy savings in digital circuits." *Id.*, 5. *Moons* teaches that typical "ConvNets run on high precision machines using 32-bit floating point number[s][,]" but "such high precision is not always necessary[.]" *Id.*, 3. *Moons* therefore reduces precision by reducing the number of bits used for the inputs (e.g., image data pixel values) and the kernel weights. *Moons* teaches that this bit reduction can be applied either by "[u]niform rescaling: all inputs and weights are scaled with the same value" or "[p]er-layer rescaling: inputs and weights are scaled on a per-layer basis." *Id.*, 3. *Moons* goal is to save energy while minimizing accuracy loss. *Id.*, 1 ("we can gain energy…up to 30x without losing classification accuracy and more than 100x at 99% classification accuracy, compared to the commonly used 16-bit fixed point number format"). This accuracy / energy tradeoff is depicted below in Figure 2b:



Id., Fig. 2b (showing accuracy as a function of the bit width of the input and weight for per layer rescaling).

Moons teaches scaling down precision for three different datasets while maintaining sufficient accuracy:



Id., Figs. 4a-4c. For LeNet, precision is scaled to 1-6 bits; for CifarQuick, precision is scaled to 5-8 bits; and for AlexNet, precision is scaled to 5-9 bits. *Id*. *Moons* therefore computes approximate convolution values "based on a first set of most significant bits"—i.e., by performing lower precision convolutions with quantized input data and weights containing the most significant bits that allot for sufficient accuracy. *Id*., 5 (noting in precision scaling, the "MSB-bit should always be placed at the MSB position"); *Dec.*, ¶¶78-79.

a) Motivation to Combine Ujiie and Moons

A POSITA would have been motivated and found it obvious to compute *Ujiie's* approximate convolutions (i.e., AppConv) using *Moons's* precision scaling approach. *Dec.*, ¶¶80-94. Namely, a POSITA would have been motivated to replace

Ujiie's Sign Connect approximations (i.e., using a single signed bit for kernel data and full bit-resolution for image data) with *Moons's* precision scaling quantization techniques (i.e., reduced bit-resolution for both kernel data and image data) to increase system accuracy. *Dec.*, ¶80. *Ujiie's* Sign Connect computations heavily approximate the kernel value by adjusting all values to simply +1 or -1. *Ujiie*, 2, 4; *see also*, *Courbariaux* (Ex. 1011), 2 (discussing "sign function" *Ujiie* explains is equivalent to Sign Connect). In other words, *Ujiie* approximates convolutions by using fast logic that cancels out the amplitude of the weight value. *Dec.*, ¶81 (describing how larger magnitude weight values lose their amplitude by being converted to 1). Indeed, as *Ujiie* teaches, using Sign Connect approximation introduces errors into the process. For example, *Ujiie* describes how binarization techniques "show[] the performance degradation due to the prediction inaccuracy...for the complex data such as color images." *Ujiie*, 6. Sign Connect specifically led to a decrease in accuracy for the CIFAR-10 dataset. *Ujiie*, Table 3 (showing an LCP accuracy of 0.860 compared to an LCP + Sign Connect accuracy of 0.849). A POSITA would have been motivated to strike the proper balance by decreasing the inaccuracies caused by Sign Connect, particularly for complex image data using a more precise approximation method while maintaining computational efficiency. *Dec.*, ¶¶82-90.

Moons's approximations use multi-bit quantization and therefore do not completely remove the weight value's amplitude. Instead, *Moons's* process performs lower bit-width approximate calculations on all weights and input data, thereby maintaining the most significant portions of the value's amplitude. *Dec.*, ¶83 (explaining that the most significant bits represent the most significant portion of the value). *Moons's* process thus offers higher accuracy than *Ujiie's* Sign Connect process while still achieving significant compression, i.e., improved computation efficiency. *Dec.*, ¶83. Accordingly, a POSITA would have understood that modifying *Ujiie* with *Moons's* quantization techniques would have enhanced the next max pooling step in *Ujiie's* process. *Dec.*, ¶84. Enhancing the accuracy of the AppConv computations allows *Ujiie's* system to better identify the most significant input data during max pooling. *Dec.*, ¶85. As such, using quantization for *Ujiie's* approximate convolution step would have decreased the likelihood of error at the max pooling stage, thereby providing more accurate decisions on the most significant dataset to perform the computationally expensive full-precision convolution. *Dec.*, ¶85. This modification would therefore enable a POSITA to take advantage of improved precision when appropriate. *Dec.*, ¶85.

Moreover, quantization offers more flexibility compared to a limited set of fixed weight values (e.g., +1 and -1). *Dec.*, ¶86. Using fixed weights across all layers prevents implementing different precision at different layers that may be more or

less tolerant to quantization. *Dec.*, ¶87. *Moons* expressly teaches "per-layer quantization" that finds "an optimal [precision] setting for each layer by exploiting the variations of the layer-specific input and weight distributions." *Moons*, 4. Replacing fixed weights with quantized representations provides layer-wise tuning, allotting for differing precisions across layers and providing a more robust CNN capable of "control[ling] the energy-accuracy trade-off tightly[.]" *Moons*, 4; *Dec.*, ¶87. Indeed, *Ujiie's* system, similar to *Moons*, would have benefitted from per-layer quantization at the approximate convolution stage to increase efficiency by using the proper amount of bits to increase the likelihood of identifying a maximum value that accurately identifies the most significant dataset, which should be processed in the full resolution convolution. *Dec.*, ¶88.

While modifying *Ujiie* to perform multiplications of low bit-width may lead to an increase in energy consumption, a POSITA would have recognized that, in some circumstances, the tradeoff for computational energy would be a desirable tradeoff for the enhanced accuracy *Moons's* process realizes. *Dec.*, ¶89. For example, in scenarios involving more complex image analysis (e.g., "color images"), a POSITA would have recognized that improved image analysis using *Moons's* quantization would have benefitted the system and would have been willing to consume additional energy to obtain this improved image analysis. *Dec.*, ¶89 (explaining that the proposed combination presents a straightforward trade off

between accuracy and energy consumption). Notably, this modification preserves *Ujiie's* goal of decreasing energy consumption in comparison to a standard CNN approach in which no approximation step is implemented. Indeed, both *Moons* and *Ujiie* describe approximation techniques that expressly seek to decrease energy consumption, with *Moons's* approach allowing for increased accuracy, as necessary. *Moons*, Figure 5(b) (showing decreased energy consumption using decreased bit width (i.e., quantization)), 2 ("Approximate computing through precision scaling lowers the active power consumption"), 3 ("To reduce the energy consumption of the ConvNet's computations, our main strategy is to quantize its weights and the inputs to its layers"), 6 ("If per-layer quantization is used…more energy can be gained, while losing less classification accuracy in the process"). Because *Moons's* approximation still significantly decreases energy consumption compared to typical full precision convolution, a POSITA would have appreciated that *Ujiie's* modified system would still gain significant improvements in energy consumption, when compared with the traditional convolutional process, while yielding higher accuracy in Step 1. *Dec.*, ¶90.

A POSITA would have appreciated a reasonable expectation of success modifying *Ujiie's* approximate convolution process with *Moons's* approximated quantization computations because it simply requires replacing one known approximation technique with another to improve *Ujiie's* process. *Dec.*, ¶¶91-94.

Both *Ujiie* and *Moons* teach approximate CNN computations for improving energy efficiency. *Compare*, *Ujiie*, 6 (teaching its LCP algorithm "reduce[s] the energy consumption for image recognition by 11.4%–14.1%), *with Moons*, Figure 6(a) (showing a significant reduction in relative energy usage as a function of bit width), 6 (teaching that "applying approximate computing" "drastically reduce[s]" "the energy consumption"). *Dec.*, ¶91. This modification would have only required minor hardware and software architecture changes to implement another known approximation technique into *Ujiie's* system to reap the benefits described above. *Dec.*, ¶92.

Moreover, a POSITA would have recognized that *Ujiie's* "multiplications of the sign of weights" would have been readily replaced by *Moons's* minimum bit-width (i.e., MSB) computations at the AppConv step. *Dec.*, ¶93. As indicated by *Moons*, "[l]ower precision computing, i.e. computing using less bits…can be applied to a common digital multiplier." *Moons*, 5. *Ujiie* expressly provides support that "general multipliers" are typically used in the computation process. *Ujiie*, 4. Thus, it would have been well within the skill of a POSITA to incorporate well-known techniques, like *Moons's* "[l]ower precision computing," in *Ujiie's* system to perform computations using the MSBs from the input data and weights. *Dec.*, ¶93.

Additionally, *Ujiie's* hardware would allow for use of the MSBs in the approximate computation while retaining the full value in memory for use in the full

precision computation once a max value has been determined. *Dec.*, ¶94. For example, *Ujiie* teaches that the results of the AppConv and pooling are "temporarily stored in the index memory connected to the pooling circuit and utilized as the signal that control inputs to the Conv circuit." *Ujiie*, 4-5. This is shown in Figure 4 below:

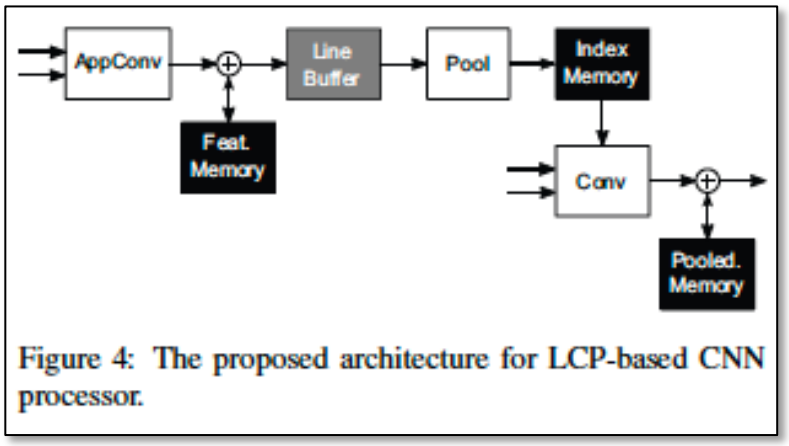


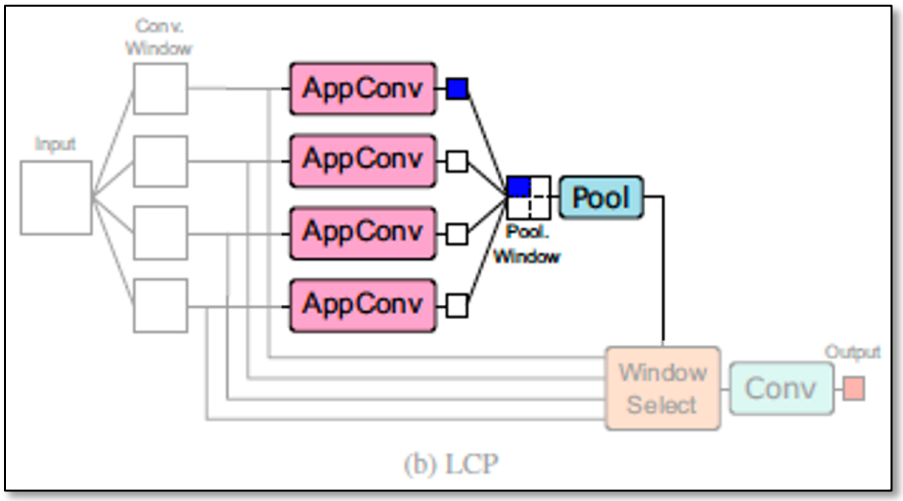
Figure 4: The proposed architecture for LCP-based CNN processor.

Id., Fig. 4. In the modified system, the approximate computations performed on the MSBs of the input data and weights would similarly be stored in the index memory. *Dec.*, ¶94. Then, full precision is performed on the dataset that exhibited the maximum value from the approximate computations. *Dec.*, ¶94.

3. *Claim 1(b): "examining a first set of values computed for the plurality of data sets in the first iteration to determine whether a maximum value is present among the first set of values;"*

Ujiie-Moons teaches computing a first set of values for the plurality of datasets in the first iteration. *Supra*, Claim 1(a). *Ujiie's* system then examines the first set of values to determine whether a maximum value is present among the first set of

values in a "max pooling operation[.]" *Ujiie*, 1. *Ujiie's* approximate max pooling operation is illustrated in Figure 2b below:



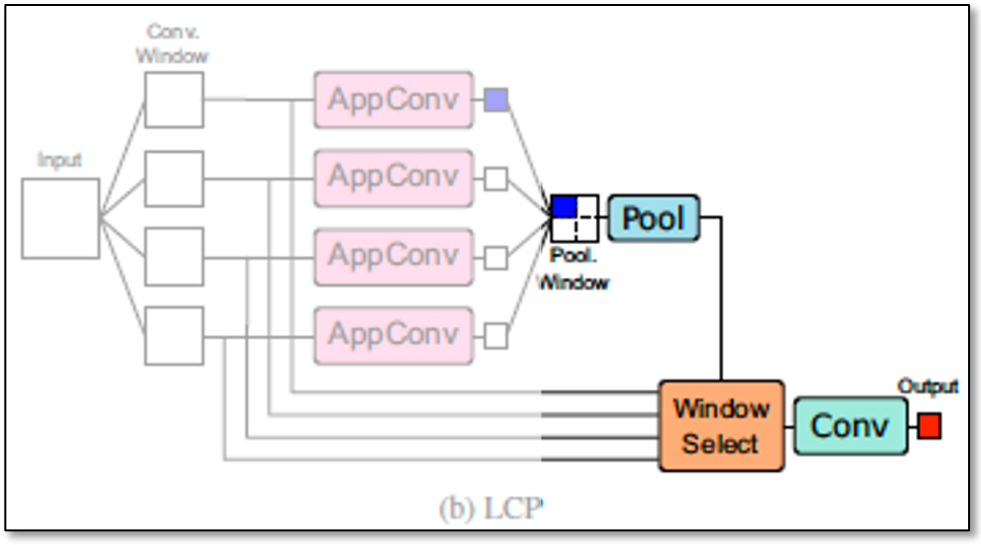
Id., Fig. 2b (modified for emphasis).

Ujiie teaches that it "use[s] max pooling, which selects the max value from the pooling window." *Ujiie*, 2. In *Ujiie's* LCP computation flow, "Step 1" computes "AppConv using the lightweight and approximate computation"—which, as modified by *Moons* calculates AppConv using precision scaling. *Id.*, 4. Next, *Ujiie* proceeds to "Step 2" to "[o]perate max pooling to the results of AppConv to specify the convolution index of which the convolution result is propagated." *Id.* In Figure 2b, *Ujiie* calculates four approximate AppConv results, examines those results using max pooling, and the largest value is determined as the maximum value, shown as a dark blue pixel. *Dec.*, ¶¶95-96 (discussing *Ujiie*, 2 and 3). In this way, *Ujiie* "**adopt[s] the approximation only to predict the region to be selected in the max**

pooling" (i.e., *determine[s] whether a maximum value is present among the first set of values*). *Ujiie*, 3.

4. Claim 1(c): "responsive to identifying the maximum value, performing a full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value; and"

Ujiie teaches the goal of its LCP computing method is to "predict[] a local region that will be **selected later in the pooling layer to limit the application of the accurate and thus expensive convolutional operations to the predicted region.**" *Ujiie*, 1. In other words, *Ujiie* limits the expensive full precision computations to the dataset identified as a maximum from the max pooling step. *Ujiie*, 4; *Dec.*, ¶97. Indeed, in Step 3 of *Ujiie's* LCP computing method, responsive to identifying the maximum, *Ujiie* "[c]ompute[s] Conv only to the input corresponding to the specified index in Step 2." *Ujiie*, 4. This "Conv" computation is a "regular (and exact) convolution unit (Conv)[.]" *Id*. This exact (full precision) convolution is shown below in Figure 2b:

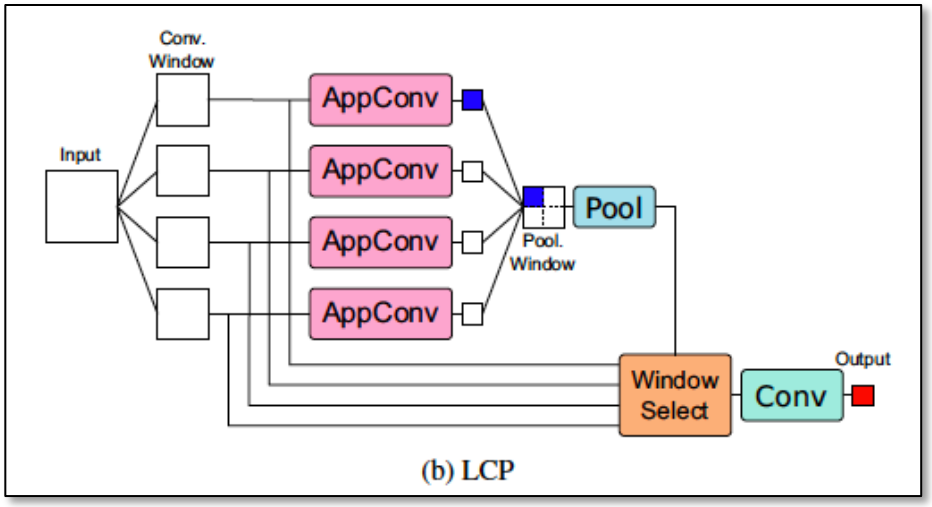


Id., Fig. 2(b) (modified for emphasis).

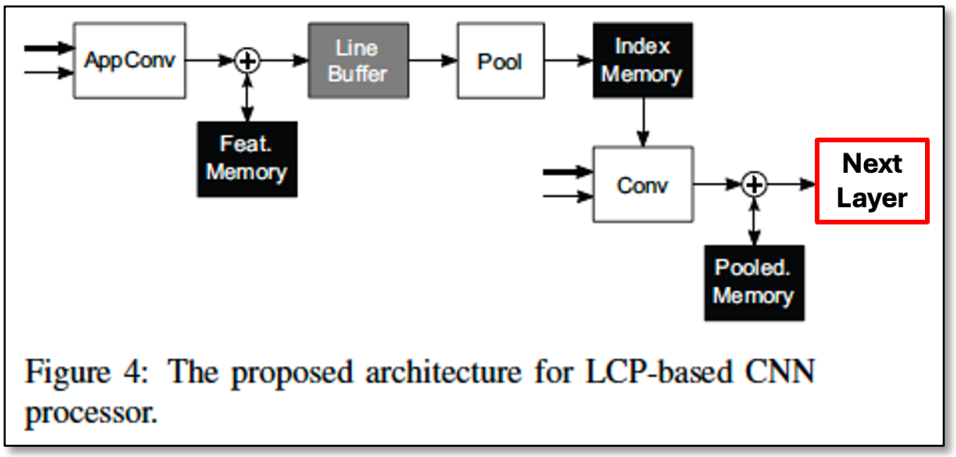
Ujiie's "regular" "exact" Conv convolution is a "full precision" convolution. *Dec.*, ¶98. Regular, exact convolutions use all available bits (full precision) and are accurate, but computationally expensive. *Id*. That is why *Ujiie's* goal is to limit the number of exact, full precision convolutions to the most important datasets exhibiting the max from the Step 2 max pooling operation. *Id*.

5. Claim 1(d): "propagating the full precision computation of the value to a subsequent layer of the CNN."

Once *Ujiie's* system computes a full precision computation on the dataset that exhibited the maximum, that full precision value is propagated as output to the next layer. *Ujiie*, 3; *Dec.*, ¶99. This full precision output is shown in Figure 2b below as a red pixel:



Id., Fig. 2b. *Ujiie* also provides an overview of its proposed architecture in Figure 4:



Id., Fig. 4 (annotated). The final arrow shows that the Conv results stored in pooled memory are sent to a subsequent layer. *Dec.*, ¶100. This would include another convolution and pooling layer, and eventually to a fully connected layer. *Dec.*, ¶¶100-101.

Ujiie exemplifies this process in Figure 1, showing how in typical CNNs, the output of convolution/pooling is propagated to "Layer 2" of the CNN for additional convolution and pooling steps or to a "Fully Connected" layer.

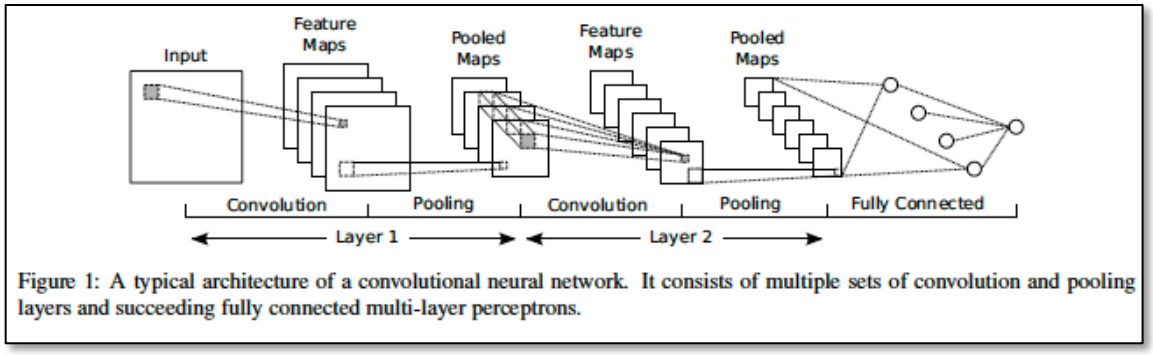


Figure 1: A typical architecture of a convolutional neural network. It consists of multiple sets of convolution and pooling layers and succeeding fully connected multi-layer perceptrons.

Ujiie, Figure 1; *see also id*., 3 ("Figure 1: A typical architecture of a convolutional neural network. It consists of multiple sets of convolution and pooling layers and succeeding fully connected multi-layer perceptrons"). Further, *Ujiie* provides Tables 1 and 2 that show multiple convolution and pooling and fully connected layers. *Id*., 5. Therefore, *Ujiie's* system propagates the full precision computation of the value (i.e., exact convolution output) to a subsequent layer of the CNN—e.g., "Layer 2" or the Fully Connected layer of *Ujiie's* CNN. *Dec.*, ¶¶101-102.

B. Claim 10 – The one or more non-transitory computer-readable storage media of claim 1, wherein the CNN is employed to analyze an image.

Ujiie teaches the one or more non-transitory computer-readable storage media of claim 1. *Supra*, Claim 1.

Ujiie teaches "[t]his paper proposes an efficient computing method, LazyConvPool (LCP), and its hardware architecture to reduce power consumption of **CNN-based image recognition**." *Ujiie*, Abstract. *Ujiie* proposes an "experimental evaluation using **image classification dataset**[**s**]" such that its approximated prediction strategy for CNNs reduces power consumption for

analyzing image data. *Id*.; *see also id*., 5 ("We evaluated the classification accuracy of LCP with image recognition datasets. We used two datasets: the handwritten digits dataset MNIST [] and the object recognition dataset CIFAR-10"); Table 1 ("input image"), Table 2 ("input image"). *Ujiie* teaches that through "the experimental evaluation, the proposed LazyConvPool algorithm and its hardware architecture successfully reduce the energy consumption for **image recognition** by 11.4%–14.1% while keeping the recognition accuracy." *Ujiie*, 6.

C. Claim 11 – The one or more non-transitory computer-readable storage media of claim 1, wherein:

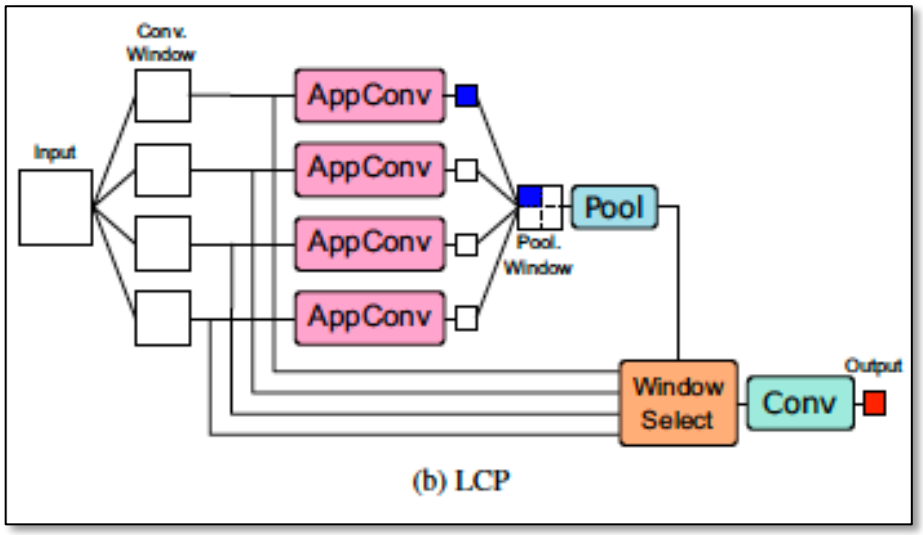
Ujiie teaches the one or more non-transitory computer-readable storage media of claim 1. *Supra*, Claim 1.

1. Claim 11(a): "the first iteration computes a value that approximates the full precision computation of the value; and"

Ujiie teaches computing a value via AppConv that approximates the full precision computation of the value. *Supra*, Claim 1(a); *see also Ujiie*, 4 ("we construct LCP framework with two main parts: **(1) approximated convolution** units (AppConv) **to predict the feature selection** in the pooling layer, and (2) **a single regular (and exact) convolution** unit (Conv) to perform convolution **only on the predicted window**."). As modified by *Moons*, this approximation involves quantizing the input data and weights. *Supra*, Claim 1(a).

2. Claim 11(b): "the full precision computation is performed on the data set that includes less data than the plurality of data sets."

Ujiie's system only performs full precision computations on the dataset whose approximated value exhibits the maximum. *Supra*, Claims 1(b)-(c); *Ujiie*, 3 ("**LCP only executes an exact convolution (Conv) against the specified window predicted by the approximate convolution (AppConv)**."). By limiting the full precision computation to the dataset exhibiting the maximum approximated value, the full precision computation is performed on less data than the plurality of datasets. *Dec.*, ¶107. Said another way, the full precision computation is performed on the dataset that includes less data than the plurality of datasets because, in its example, *Ujiie* does not perform full precision computation on three of the four datasets but rather only on the maximum approximated value. *Id*. This maximum approximated value is indicated by the dark blue pixel in Figure 2b:



Ujiie, Fig. 2b; *Dec.*, ¶107.

VI. GROUNDS 2 AND 3: CLAIMS 2-6, 8-9 ARE OBVIOUS OVER UJIIE, MOONS, AND KAUL, OR ALTERNATIVELY, CLAIMS 2-6, 8-9 ARE OBVIOUS OVER UJIIE, MOONS, KAUL, AND ISHII

A. Claim 2 – The one or more non-transitory computer-readable storage media of claim 1, further comprising: responsive to determining that the first set of values are the same, performing, by the at least one processor, a second iteration that includes computing the value based on a second set of MSBs for each of the plurality of data sets, the second set of MSBs being larger than the first set of MSBs.

Ujiie teaches the one or more non-transitory computer-readable storage media of claim 1. *Supra*, Claim 1.

1. The '831 Inventors Cited to Kaul's Adaptive Precision Process in a Related Publication, Conceding That Iteratively Increasing Bit-Widths in Neural Networks was not new

The concept of iteratively increasing bit-widths in neural network computations is not new. The inventors of the '831 Patent published a paper in 2021 that overlaps with the disclosures in the '831 Patent. *Compare Kim*, (Ex. 1012), Abstract (proposing "*precision cascading*, where the input features are divided into a number of low-precision groups and approximate convolutions with only the most significant bits (MSBs) are performed first. Based on this approximate computation, the full-precision convolution is performed only on the maximum pooling output"), *with '831 Patent*, Abstract ("the input features may be divided into a group of precision values and the operation(s) may be cascaded. A maximum may be

identified…using a small number of bits in the input features, and the full-precision convolution may then be performed on the maximum input.").

In *Kim*, the inventors explain that precision cascading derived from "[a]daptive precision," a concept that "has been presented" previously, citing to *Kaul*. *Kim*, 2. *Kaul's* "adaptive precision" referenced by the inventors involves using "as-needed precision" that "start[s] with low-accuracy from the MSB[.]" *Kaul*, 1. *Kaul's* process then "advances MSB to LSB" "with a successive 2b [2 bits]…in each iteration" until the k-nearest-neighbor is found. *Id.* This process "improves energy efficiency" of "k-nearest-neighbor (kNN) computations [which] are key building blocks for computer vision, classification, and machine-learning workloads[.]" *Id.*

The inventors of the '831 Patent, well aware of *Kaul's* "adaptive precision" technique used for "computer vision," admittedly applied this concept to CNNs (i.e., a "building block[] for computer vision"), and called it "precision-cascading[.]" *Kim*, 2 (admitting "[a]daptive precision scheme has been presented in [*Kaul*] only for *k*-nearest neighbor algorithms, not for large-scale DCNNs"). Accordingly, *Kaul* is applicant-admitted art that teaches the progeny process for precision cascading (i.e., adaptive precision) in the same field (i.e., computer vision).

2. Ujiie-Moons in View of Kaul

Ujiie-Moons teaches performing an iteration that computes an approximated value based on MSBs and then uses max pooling to identify a single dataset that

exhibits the maximum to perform a full precision convolution on. *Supra*, Claim 1(a)-(c). However, *Ujiie-Moons* does not discuss what happens when max pooling does not identify a maximum. For example, when the output approximated results for two or more datasets are the same. *See, e.g.*, *Ishii* (Ex. 1007), [0130], ("Note that in a case where there are two or more maximum values, the output selecting unit 124 need just select any one of the two or more maximum values of the processing results"), [0142] (describing selecting maximum after max pooling despite a tie), Fig. 5 (depicting the same); *Dec.*, ¶112 (discussing common tie-breaking strategies in neural networks).

A POSITA would have understood in *Ujiie-Moons*, if two datasets share the same approximated value in a max pooling process, a single dataset cannot be accurately selected for a full precision convolution under *Ujiie's* native prediction strategy. *Dec.*, ¶113. A POSITA would have looked to known techniques to resolve shared approximated values following max pooling to identify a maximum enabling *Ujiie's* prediction strategy to select a single dataset representing the most significant input data for full precision convolution. *Id. Kaul* provides a known technique to solve this issue. *Kaul* teaches iteratively increasing bit width in neural network computations when a first iteration does not provide an answer.

Kaul, similar to *Ujiie* and *Moons*, teaches an "[a]daptive precision" method for computer vision that "improves energy efficiency by eliminating a majority of

vectors without costly full-precision computation, with as-needed precision refinement to guarantee…accuracy[.]" *Kaul*, 1. Specifically, *Kaul's* method "start[s] with low-accuracy from the MSB using partial distance compute circuits…[then] refine[s] the remaining distances, with a successive 2b [2 bits] of the accumulated distance available for further sorting in each iteration." *Id.* This "[i]terative refinement of accumulated distance is interleaved as needed with successive sort operations on 2b windows from MSB to LSB." *Id. Kaul's* iterative process is illustrated below:

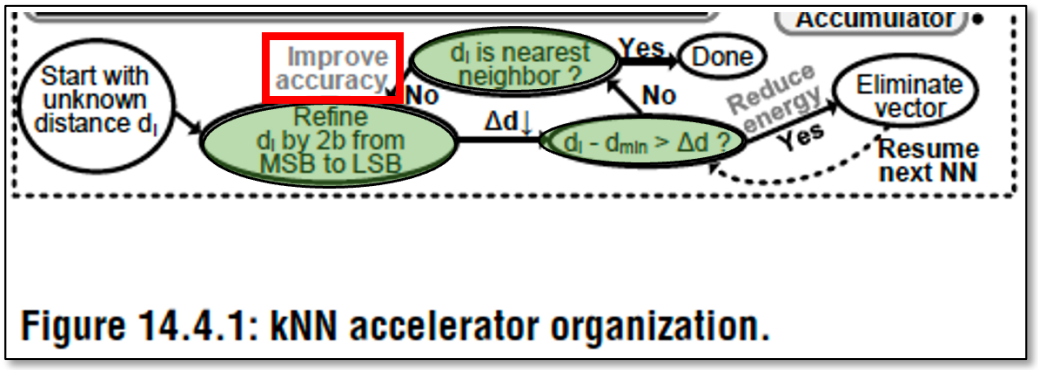


Figure 14.4.1: kNN accelerator organization.

Id., Fig. 14.4.1 (excerpted and annotated).

As shown in Fig. 14.4.1, *Kaul's* method determines if the nearest neighbor was found using the current bit-precision, and if not, "[i]mprove[s] accuracy" by refining "by 2b [2 bits] from MSB to LSB[.]" *Id.*, Figure 14.4.1. Therefore, in *Kaul*, if the nearest neighbor is not found, *Kaul* performs a second iteration of computations that includes two additional bits. *Dec.*, ¶115.

In *Ujiie-Moons-Kaul*, *Ujiie's* modified system performs a first iteration of approximate convolutions using a first number of MSBs, performs max pooling, and if no maximum is found (i.e., *responsive to determining that the first set of values are the same*…), performs a second iteration pursuant to *Kaul* with two additional bits (i.e., "performing a second iteration that includes computing the value based on a second set of MSBs for each of the plurality of data sets, the second set of MSBs being larger than the first set of MSBs"). *Dec.*, ¶¶113, 116.

a) Motivation to Combine *Ujiie*, *Moons*, and *Kaul*

A POSITA would have understood that by using decreased approximated precision before max pooling, as taught by *Ujiie-Moons*, a clear maximum may not always be determined. *Dec.*, ¶117. This would occur, for example, when analyzing an area of the image with minimal differences with the surrounding area. *Dec.*, ¶117 (describing how encountering subsets of images with similar pixel input data is commonplace in image analysis). A POSITA would have understood that with no maximum—such as a tie described in *Ishii*—it would have been obvious in view of *Kaul* to iterate by computing a higher precision approximation to distinguish between the similar pixel input data to identify which, if any, produces a maximum value. *Dec.*, ¶118.

This would have increased the accuracy of the AppConv process without substantially increasing energy consumption. *Dec.*, ¶¶119-120; *Kaul*, 1 ("Adaptive

precision improves energy efficiency…with as-needed precision refinement to guarantee kNN accuracy of closely matched vectors"), *see also id*. (describing shorter bit-width also "reduces…latency[,]" decreases storage area usage, and "reduces [energy usage] up to 3x compared to a full-precision design"). For example, iterating with two more precise bits would support a more precise output value while striking a balance for energy expenditure, latency, and storage. *Dec.*, ¶119. Moreover, this would have ensured *Ujiie's* native prediction strategy would find a single max dataset, performing iterations until a delta between datasets is met exhibiting a maximum. *Dec.*, ¶120.

A POSITA would have appreciated a reasonable expectation of success modifying *Ujiie-Moons* to perform *Kaul's* adaptive precision technique. *Dec.*, ¶¶121-125. *Ujiie's* annotated Figure 4 below shows an example of how this process would work:

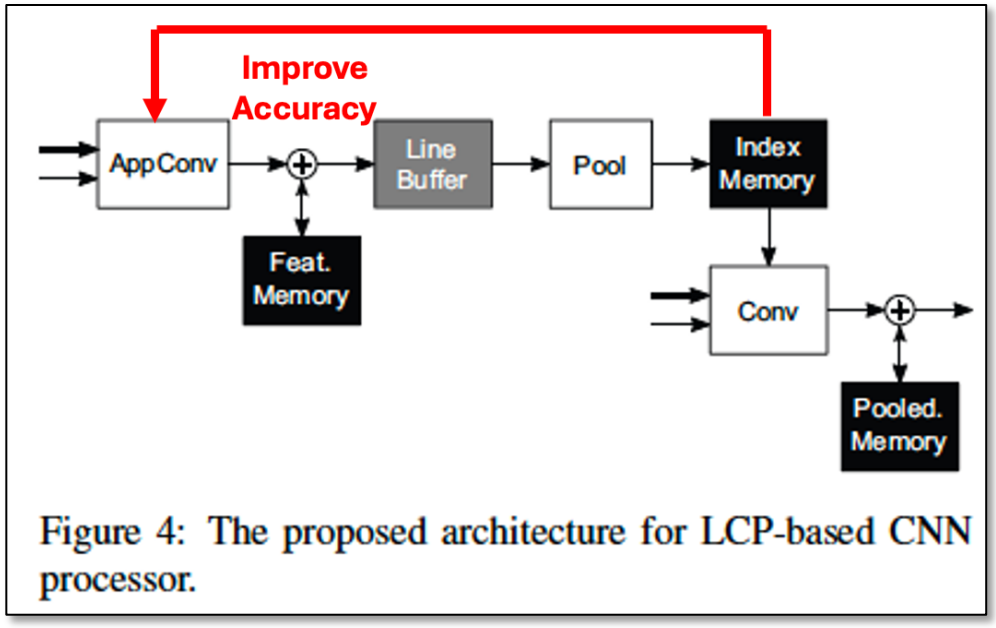


Figure 4: The proposed architecture for LCP-based CNN processor.

Ujiie, Fig. 4 (annotated).

As shown, *Ujiie-Moons's* system performs a normal iteration of low bit precision AppConv and max pooling. *Dec.*, ¶122. The results of the pooling operation are accumulated in Index Memory, and the results are analyzed for the maximum to determine which dataset receives full-precision Conv. *Dec.*, ¶122. If no maximum value exists (e.g., two datasets tied) on the first iteration, the process proceeds back to AppConv for a second iteration using an increased number of MSBs, as taught by *Kaul*. *Kaul*, 1, Fig. 14.4.1; *Dec.*, ¶¶122-123. The system would then perform the second iteration of convolution and max pooling with the weights and input containing more MSBs than the first iteration. *Dec.*, ¶123. *Ujiie's* memory would allow for retrieval of the original data in its full precision format to allow for modification of the bit width as taught by *Moons*. *Dec.*, ¶123.

A POSITA would have understood that *Kaul's* kNN-based adaptive precision technique is compatible with *Ujiie's* CNN processing. *Dec.*, ¶¶124-125. For example, convolution computations in CNNs and kNN vector comparisons both involve multiplication and addition across vectors. *Dec.*, ¶124. In this case, the precision of the values within the vector (i.e., each value's bit width) would affect the computations in essentially the same way. *Dec.*, ¶125. As such, the same bit width implications for computational efficiency would apply to both. *Dec.*, ¶125. Specifically for the present analysis, both would operate at lower bit width values in the vectors for the initial computations and then, when a resolution is undeterminable (e.g., no maximum value in CNNs or no nearest neighbor in kNNs), increase precision by including more bits in the vector values. *Dec.*, ¶125. Therefore, a POSITA would have realized that adaptive precision would be readily applied to vector-based computer vision computations, including in both CNNs and kNNs. *Dec.*, ¶125.

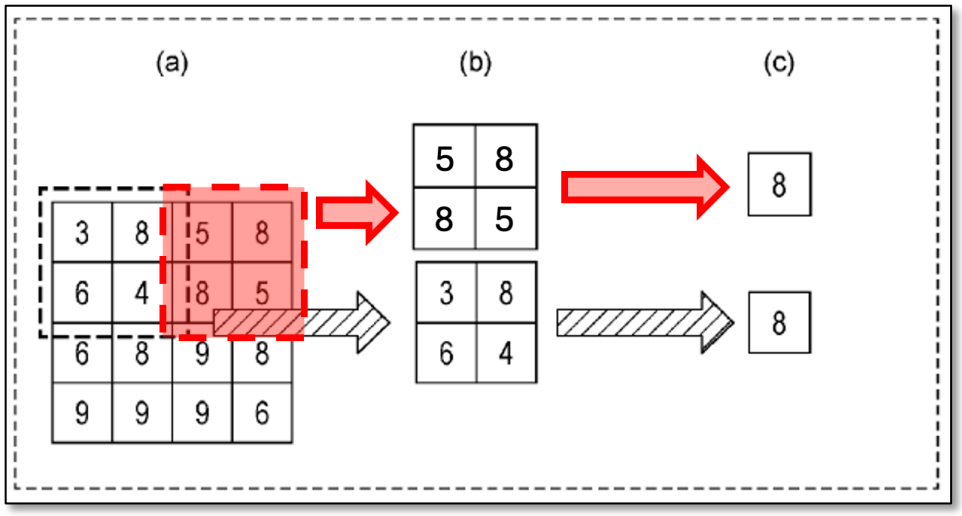
3. Ujiie-Moons-Kaul in View of Ishii

Ujiie-Moons-Kaul teaches a process to identify the single dataset that receives a full precision convolution when two or more datasets share the same approximated convolution value by iteratively increasing bit-widths for the approximated computations. *Supra*, §VI.A.2. *Ujiie's* native prediction strategy requires identifying a maximum. *Dec.*, ¶126. Therefore, *Ujiie's* process necessarily determines if the

approximated values exhibit a maximum, or are the same and thus fails to identify a maximum. *Dec.*, ¶126. *Ujiie's* modified system therefore necessarily performs a computational check whether the values are the same. *Dec.*, ¶126. This computational check was well-known in the art, as evidenced by *Ishii* above. *Dec.*, ¶126. To the extent Patent Owner argues that *Ujiie* does not expressly describe checking if the output of its max pooling process identifies a tie and that *Ishii's* teachings need to be a formal component of Petitioner's challenge, it would have been obvious to incorporate *Ishii's* express teachings into *Ujiie's* modified system. *Dec.*, ¶¶127-133.

Ishii teaches a "classifier 90a includes an image input unit 91 that reads an image (input image), a convolution processing unit 92 that performs convolution processing on the input image, [and] a subsampling unit 93 that performs subsampling [i.e., max-pooling] processing[.]" *Ishii*, [0059], [0073] ("selecting a maximum value by subsampling is called Max-pooling"). *Ishii* demonstrates a scenario in which the results of max-pooling include a tie between two values. For example, *Ishii* teaches the subsampling processing includes sampling a 2x2 region "for selecting a maximum pixel value[.]" *Id.*, [0142]. *Ishii's* examples include both a scenario in which the maximum is straightforward (i.e., between "pixel values 3, 8, 6, and 4") and scenarios in which there is a tie (e.g., in the "2x2 region including

an upper right pixel" as shown in Figure 5, showing values 5, 8, 8, and 5). *Id.* Figure 5 is reproduced below:



Id., Fig. 5 (annotated).

As shown, two or more values exhibit the maximum (e.g., the value "8" here). *Dec.*, ¶¶127-128. *Ishii* teaches or at least suggests that its system would choose between one of the two values exhibiting the maximum, such that the system "selects 8…as representative…[of the] 2x2 region including [the] upper right pixel[.]" *Ishii*, [0142]; *Dec.*, ¶128. *Ishii* thus teaches or suggests a computational check to identify a maximum, such that if there is a tie, *Ishii's* system selects one to proceed to the next layer. *Dec.*, ¶¶127-128.

Indeed, *Ishii* teaches a similar selection process for its convolutional computations, where *Ishii* "selects a maximum value from among values…of three processing results of the convolution processing" including results from "low-resolution convolution[,]" "medium-resolution convolution[,]" and "high-resolution

convolution[.]" *Ishii*, [0119], Fig. 4. In determining its maximum here, *Ishii* teaches "in a case where there are two or more maximum values, the output selecting unit 124 need just select any one of the two or more maximum values of the processing results." *Id.*, [0130]. A POSITA would have understood or at least found obvious that this same tie-break process for same-value convolutions would have been used to tie-break *Ishii's* max pooling process (i.e., the processor "select[s] any one of the two or more maximum values"). *Dec.*, ¶129.

a) Motivation to Combine *Ujiie*, *Moons*, *Kaul*, and *Ishii*

A POSITA would have been motivated and found it obvious to modify *Ujiie-Moons-Kaul* to perform computational checks on the maximum value of *Ujiie's* max pooling operation to determine if two or more values exhibited the maximum, as taught by *Ishii*. *Dec.*, ¶¶130-133. Namely, a POSITA would have understood the low precision approximate computations in *Ujiie's* modified system would cause scenarios where two or more values exhibit the maximum. *Dec.*, ¶130. As such, a POSITA would have been motivated to implement *Ishii's* computational check in *Ujiie's* modified system to identify this scenario and resolve it. *Dec.*, ¶130. To be clear, the proposed combination does not simply choose one of the tie value datasets as in *Ishii*. Instead, as set forth below, the combination adds bits and re-computes the AppConv values to identify which of the original datasets produces a maximum value.

Indeed, a POSITA would have understood *Ujiie's* native prediction strategy that mandates a single dataset to exhibit the max for full precision convolutions to teach, suggest, or motivate the combination with a reference that teaches a resolution if a maximum is not found, such as in *Ishii*. *Dec.*, ¶132. Using *Ishii's* random selection tie-break functionality is, however, error prone. *Dec.*, ¶132. For instance, by randomly selecting one of two choices, there's a fifty percent chance of error. *Dec.*, ¶132. A POSITA would have instead used *Kaul's* technique to increase bit-widths rather than *Ishii's* random selection to increase system accuracy and for the reasons above. *Dec.*, ¶132.

A POSITA would have appreciated a reasonable expectation of success modifying *Ujiie-Moons-Kaul* to check whether two or more values exhibit the maximum. *Dec.*, ¶133. *Ujiie's* modified system already processes approximate convolution results and stores them in its "Index Memory." *Ujiie*, 4-5, Fig. 4. Thus, this modification would have only required minor software changes in *Ujiie's* system to check for a tie between these values for the benefits described above. *Dec.*, ¶133.

B. Claim 3

1. Claim 3(a): "The one or more non-transitory computer-readable storage media of claim 2, further storing instructions, which, when executed by the at least one processor, cause the at least one processor to perform operations comprising: examining a second set of values computed for the plurality of data sets in the second iteration to determine whether the maximum value is present among the second set of values; and"

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 2. *Supra*, Claim 2.

Ujiie's modified system stores instructions executed by its processor that perform a second iteration that computes a second set of AppConv values based on a second set of MSBs for each of the plurality of datasets. *Supra*, Claim 2. *Ujiie's* modified system further examines a set of values computed for the plurality of datasets in the first iteration to determine whether a maximum value is present among the set of values. *Supra*, Claim 1(b). A POSITA would have understood that *Ujiie's* modified system would analyze the second set of values computed in the second iteration to determine whether a maximum value is present in the same way as it analyzes the first set of values. *Dec.*, ¶¶95-96, 135-136.

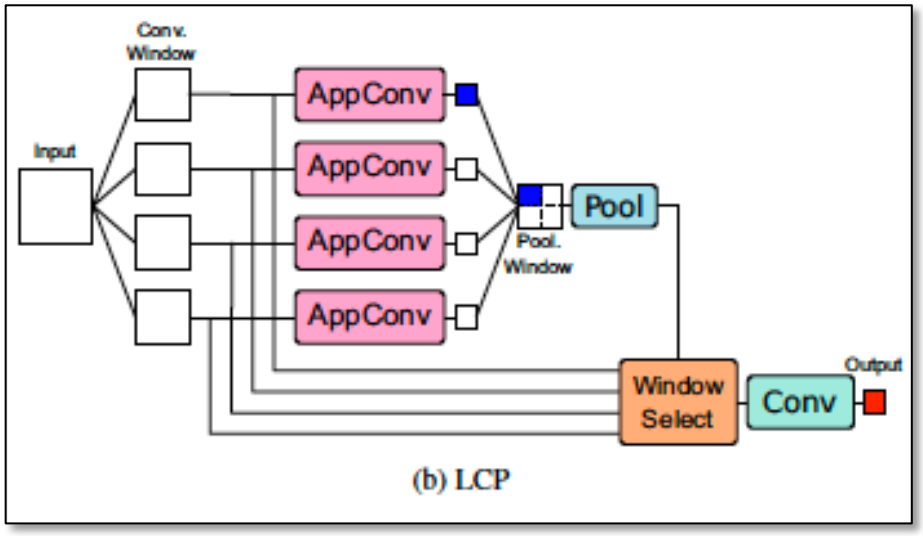
2. Claim 3(b): "responsive to identifying the maximum value among the second set of values, performing, by the at least one processor, the full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value in the second iteration."

At stated above, *Ujiie-Moons-Kaul* (Ground 2) or alternatively, *Ujiie-Moons-Kaul-Ishii* teaches performing a full precision computation on the dataset that exhibited the maximum value, including in a second iteration with increased bit-precision (i.e., "responsive to identifying the maximum value among the second set of values, performing, by the at least one processor, the full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value in the second iteration"). *Supra*, Claim 2; *see also* Claim 1(c) (describing similar process for the first iteration). A POSITA would have understood, as explained above, that *Ujiie's* modified system would perform a full precision computation of the value for a dataset, of the plurality of datasets, that exhibited the maximum value in the second iteration in the same way that it performs a full precision computation for the first iteration. *Dec.*, ¶¶97-98, 137-138.

C. Claim 4 – The one or more non-transitory computer-readable storage media of claim 2, wherein the computing in each of the first iteration and the second iteration employs a convolution and a pooling.

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 2. *Supra*, Claim 2.

Ujiie teaches performing convolution (e.g., approximate convolution "AppConv" and/or exact convolution "Conv") and pooling (i.e., max pooling) in both the first iteration and the second iteration. *Supra*, Claims 1(a)-(c), 2, 3; *see also*, *Ujiie*, 4 ("Figure 2b illustrates the LCP-based convolution and pooling operations."), Figure 2b:



Id., Fig. 2b.

D. Claim 5 – The one or more non-transitory computer-readable storage media of claim 4, wherein the convolution is a NxN convolution, where N is any integer.

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 4. *Supra*, Claim 4.

Ujiie teaches specific network parameters, such as the size of its convolution filters, used for its LCP process on different datasets, including MNIST and CIFAR-10. *Ujiie*, 5, Tables 1-2. In these network parameters, *Ujiie* specifies that it uses a

"**Convolution**" "kernel" that is "**5x5**" where N in this case is equal to 5. *Id.* Tables 1 and 2 showing the 5x5 convolutions are shown below:

Layer name	Parameter
Input image	size: 28x28, channel: 1
Convolution	kernel: 5x5, channel: 20
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 50
Max pooling	kernel: 2x2, stride: 2
ReLU	
Fully connected	channel: 500
ReLU	
Fully connected	channel: 10
Softmax	

Table 1: Network parameters of CNN for MNIST dataset.

Layer name	Parameter
Input image	size: 32x32, channel: 3
Convolution	kernel: 5x5, channel: 64, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 128, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 256, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 512, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 512, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Fully connected	channel: 512
ReLU	
Fully connected	channel: 10
Softmax	

Table 2: Network parameters of CNN for CIFAR-10 dataset.

Id., Tables 1 and 2 (annotated).

E. Claim 6 – The one or more non-transitory computer-readable storage media of claim 4, wherein the pooling is a NxN pooling, where N is any integer.

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 4. *Supra*, Claim 4.

Ujiie teaches other specific network parameters, such as the size of the max pooling window, used for its LCP process on different datasets, including MNIST and CIFAR-10. *Ujiie*, 5, Tables 1-2. In these network parameters, *Ujiie* specifies that it uses a "**Max pooling**" "kernel" that is "**2x2**" where N in this case is equal to 2. *Id.* Tables 1 and 2 showing the 2x2 max pooling windows are shown below:

Layer name	Parameter
Input image	size: 28x28, channel: 1
Convolution	kernel: 5x5, channel: 20
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 50
Max pooling	kernel: 2x2, stride: 2
ReLU	
Fully connected	channel: 500
ReLU	
Fully connected	channel: 10
Softmax	

Table 1: Network parameters of CNN for MNIST dataset.

Layer name	Parameter
Input image	size: 32x32, channel: 3
Convolution	kernel: 5x5, channel: 64, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 128, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 256, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 512, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Convolution	kernel: 5x5, channel: 512, padding: 2
Max pooling	kernel: 2x2, stride: 2
ReLU	
Fully connected	channel: 512
ReLU	
Fully connected	channel: 10
Softmax	

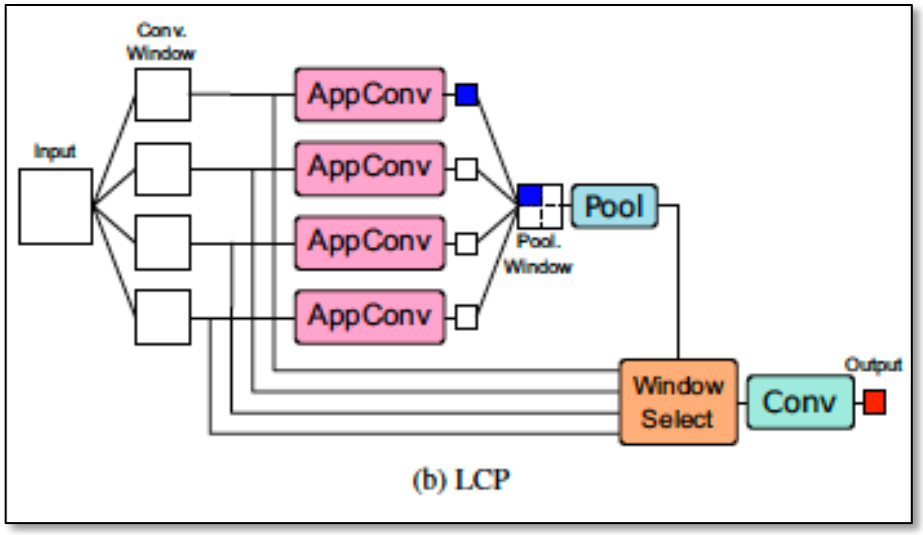
Table 2: Network parameters of CNN for CIFAR-10 dataset.

Id., Tables 1 and 2 (annotated).

F. Claim 8 – The one or more non-transitory computer-readable storage media of claim 2, wherein at least one of the first iteration and the second iteration is performed with a precision less than that of the full precision computation.

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 2. *Supra*, Claim 2.

Ujiie teaches performing an approximate convolution "AppConv" followed by an exact full precision convolution "Conv." *Supra*, Claims 1(a)-(c); *see also*, *Ujiie*, 4 (teaching the two main parts of LCP are "(1) approximated convolution units (AppConv) to predict the feature selection in the pooling layer, and (2) a single regular (and exact) convolution unit (Conv) to perform convolution only on the predicted window"). This process is reproduced below in Figure 2b:

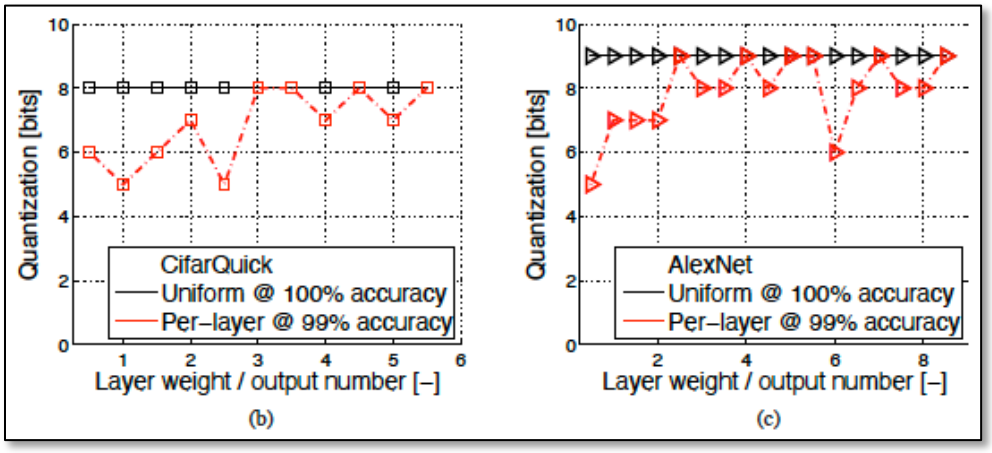


Id., Fig. 2b. As described above, it would have been obvious, and a POSITA would have been motivated to modify *Ujiie's* AppConv computations that use Sign Connect to approximate convolutions with *Moons* precision scaling to perform those approximations. *Supra*, Claims 1(a)-(c). *Moons* precision scaling approximates convolutions using less bits than the exact convolution. *Id*. *Ujiie's* modified approximated convolution is thus performed with a precision less than that of the full precision computation. *Dec.*, ¶¶145-146. Since the approximated computation is performed before the exact, full precision computation, at least the first iteration computation is performed at a lower precision than the full precision computation. *Dec.*, ¶¶145-146.

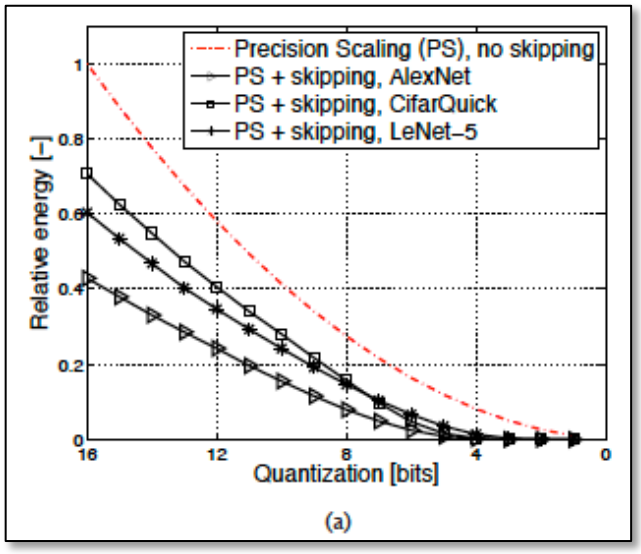
G. Claim 9 – The one or more non-transitory computer-readable storage media of claim 8, wherein the precision is 8-bit precision.

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 8. *Supra*, Claim 8.

Moons teaches its per-layer quantization uses 8-bit precision in at least some of its layers. For example, when using the CIFAR and AlexNet databases, multiple layers of convolution and pooling include 8-bit precision. *Moons*, 5, Figure 4:

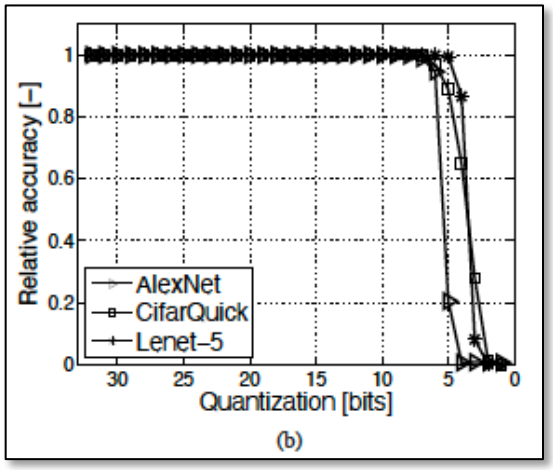


Id., Fig. 4. Furthermore, *Moons* teaches precision scaling at 8-bit precision decreases relative energy consumption by roughly 70% as compared to 16-bit precision. *Moons*, Figure 6a:



Id., Fig. 6a.

A POSITA would have been motivated and found it obvious to modify *Ujiie* to perform its approximate computations using 8-bit precision, as taught by *Moons*, for the same reasons stated above. *Supra*, Claim 1(a). For example, a POSITA would have recognized that, at least in some instances, using 8-bit precision for the approximate convolution would have provided the necessary accuracy for determining the maximum approximate value while still producing significant energy consumption savings. *Dec.*, ¶149; *see also*, *Moons*, Fig. 4(b)-(c) (showing that 8-bit precision was the necessary bit width to retain 99% accuracy). Indeed, *Moons* teaches that "[e]ach network performs at its original accuracy up to quantization with no more than 8 bits. After that, the accuracy again drops quickly." *Id.*, 4, Figure 2b:



Id., Fig. 2b. As such, a POSITA would have understood the benefits of using 8-bit precision for the approximate convolution as this would have provided significant energy savings, an express goal of *Ujiie*, while retaining sufficient precision to

determine a maximum value and subsequently perform full precision computation on the maximum value's dataset. *Dec.*, ¶150.

A POSITA would have further understood that, if using a lower bit precision in the first iteration (e.g., 6-bit precision), the second iteration would have used 8-bit precision computation as taught by *Kaul* because *Kaul* teaches MSB- to LSB-based selective distance refinement using successive 2 bits in each iteration. *Dec.*, ¶¶113-116, 151; *Supra*, Claim 2. For example, when performing a second iteration because the first 6-bit iteration did not identify a maximum value, *Ujiie-Moons* as modified by *Kaul* would have added 2 bits to the original 6-bit width, computing approximate convolutions on the 8-bit values. *Dec.*, ¶¶114-115 (discussing *Kaul's* teachings for iterating with the addition of 2 bits), 151; *Supra*, Claim 2. As such, a POSITA would have understood that *Ujiie's* second iteration would be performed in 8-bit precision, as taught by *Moons* and *Kaul*. *Dec.*, ¶¶149-151.

VII. GROUNDS 4 AND 5: CLAIM 7 IS OBVIOUS OVER UJIIE, MOONS, KAUL, AND RAVINDRAN OR ALTERNATIVELY, CLAIM 7 IS OBVIOUS OVER UJIIE, MOONS, KAUL, ISHII, AND RAVINDRAN

A. Claim 7 – The one or more non-transitory computer-readable storage media of claim 4, wherein the convolution is a 3x3 convolution, and the pooling is a 2x2 pooling.

Ujiie's modified system teaches the one or more non-transitory computer-readable storage media of claim 4. *Supra*, Claim 4.

As above, *Ujiie* teaches performing "**2x2**" "**Max pooling**." *Ujiie*, Tables 1-2; *supra*, Claim 6. In its examples, *Ujiie* teaches using a 5x5 convolution kernel. *Ujiie*, Tables 1-2; *supra*, Claim 5. *Ujiie* does not expressly discuss examples using 3x3 convolution kernels. However, varying convolution size, including 3x3 convolutions, was an extremely well-known process before the invention date of the '831 Patent. *Dec.*, ¶155; *Moons*, 6 ("**convolutional kernels** are typically very small on the order of 11 x 11, 5 x 5 or **3 x 3**."). Indeed, even the inventors of the '831 Patent describe a prior art CNN including a 3x3 convolution and 2x2 max pooling. *'831 Patent*, 5:64-6:1 ("FIG. 2 shows a flow diagram, or block diagram, of **a conventional convolution and pooling scheme that employs 3x3 convolution, 2x2 pooling**, and 8-bit precision."), Fig. 2. It would have been obvious to implement a 3x3 convolution kernel in *Ujiie's* system, as taught by *Ravindran* for the reasons explained below.

Ravindran teaches "[a]n image processing system…[that] builds and trains an ensemble of deep learning models, such as convolutional neural networks (CNNs), to accurately and automatically perform image processing to detect particular attributes of objects in a digital image[.]" *Ravindran*, [0010]. *Ravindran* teaches a series of convolutional and pooling layers, which includes "first layer [convolution] filters, each of size 5x5," and "second layer **convolutional filters**, **each of size 3x3[**.]" *Id.*, [0047]-[0048]. Furthermore, *Ravindran* teaches **convolutional "fourth**

layer filters, each of size 3x3," combined with "**max-pooling…performed over a 2x2 region**[.]" *Id.*, [0050], Fig. 6.

a) Motivation to Combine Ujiie, Moons, Ishii, Kaul, and Ravindran

A POSITA would have been motivated and found it obvious to modify *Ujiie's* modified system to use a 3x3 convolutional filter, as taught by *Ravindran*. For example, a POSITA would have understood that a 3x3 filter includes significantly less parameters (i.e., 9 parameters) than the 5x5 filter (i.e., 25 parameters) expressly taught by *Ujiie*. *Dec.*, ¶156.; *see also*, *Ujiie*, Tables 1-2 (teaching using 5x5 filters on various datasets). Decreasing the number of parameters would have provided a lighter model that is easier to train given the need to calculate/optimize 9 instead of 25 parameters. *Dec.*, ¶156. Moreover, this would have increased computational efficiency (i.e., decreased energy consumption), furthering an express goal of *Ujiie*, by providing decreased multiplication and addition steps per filter. *Dec.*, ¶157 (explaining a 3x3 filter would only require 9 multiplications and corresponding additions whereas the 5x5 filter would require 25 multiplications and corresponding additions). Overall, a POSITA would have realized the computational benefits of using a 3x3 convolutional filter would have furthered *Ujiie's* goal of "reducing power consumption of CNN processors." *Dec.*, ¶¶156-157; *Ujiie*, 3.

A POSITA would have appreciated a reasonable expectation of success replacing *Ujiie's* 5x5 convolutional filters with *Ravindran's* 3x3 filters as this would

have required simple programming techniques well within the skill of a POSITA. *Dec.*, ¶158. As mentioned above, the inventors of the '831 Patent admit that 3x3 convolutional filters were commonly used in the conventional CNN. *'831 Patent*, 5:64-6:1, Fig. 2. *Moons* confirms the same. *Moons*, 6. It would have been a routine process to adjust *Ujiie's* convolutional filter to implement other known filter sizes. *Dec.*, ¶158. Furthermore, *Ujiie's* system already utilized a convolutional filter that was 5x5. *Ujiie*, Tables 1-2. And as taught by *Ravindran*, a single system may even switch between 5x5 convolutional filters and 3x3 convolutional filters, indicating this change in filters was commonplace. *Ravindran*, [0047]-[0050], Fig. 6. As such, modifying *Ujiie's* convolution method using a 5x5 filter to use a 3x3 filter would have required simple programming techniques well known to a POSITA and resulted in an increase in computational efficiency both during training and usage of the system. *Dec.*, ¶¶156-158.

VIII. CONCLUSION

For the foregoing reasons, Petitioner respectfully requests *inter partes* review of the Challenged Claims.

Respectfully submitted,

By: /s/ Paul R. Hart
Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER

IX. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(A)(1)

A. Real Parties-In-Interest

Petitioner and CrowdStrike Holdings, Inc. are the real parties-in-interest. 37 C.F.R. § 42.8(b)(1).

B. Related Matters

The '831 Patent is presently the subject of a patent infringement lawsuit filed against Petitioner in the United States District Court for the Western District of Texas, *Skysong Innovations, LLC v. CrowdStrike, Inc. and CrowdStrike Holdings, Inc.*, Case No. 7:25-cv-00040-DC-DTG. The '831 Patent has also been asserted against Fortinet, Inc. in the United States District Court for the Eastern District of Texas, *Skysong Innovations, LLC v. Fortinet, Inc.*, Case No. 2:25-cv-00098.

C. Lead and Back-Up Counsel

Petitioner provides the following designation and service information for lead and back-up counsel. 37 C.F.R. § 42.8(b)(3) and (b)(4).

Lead Counsel	Back-Up Counsel
Paul R. Hart (Reg. No. 59,646) Paul.Hart@eriseip.com PTAB@eriseip.com Postal and Hand-Delivery Address: ERISE IP, P.A. 717 17th St., Suite 1400 Denver, Colorado 80202 Telephone: (913) 777-5600 Fax: (913) 777-5601	Adam P. Seitz (Reg. No. 52,206) Adam.Seitz @eriseip.com PTAB@eriseip.com Postal and Hand-Delivery Address: ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601
	Hunter Horton, pro hac vice Hunter.Horton@eriseip.com

	Postal and Hand-Delivery Address: ERISE IP, P.A. 7015 College Blvd., Suite 700 Overland Park, Kansas 66211 Telephone: (913) 777-5600 Fax: (913) 777-5601

D. 37 C.F.R. § 42.8(b)(4) – Service Information

Please address all correspondence to the lead and back-up counsel as shown above. Petitioner consents to electronic service by e-mail at the e-mail addresses provided above.

CLAIM LISTING APPENDIX
U.S. Patent No. 11,775,831 for Claims 1-11

Claim Designation	Claim Language
Claim 1[Pre]	One or more non-transitory computer-readable storage media storing instructions which, when executed by at least one processor, cause the at least one processor to perform operations comprising:
Claim 1[a]	in one or more layers of a convolutional neural network (CNN), performing a first iteration that includes computing a value based on a first set of most significant bits (MSBs) for each of a plurality of data sets;
Claim 1[b]	examining a first set of values computed for the plurality of data sets in the first iteration to determine whether a maximum value is present among the first set of values;
Claim 1[c]	responsive to identifying the maximum value, performing a full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value; and
Claim 1[d]	propagating the full precision computation of the value to a subsequent layer of the CNN.
Claim 2	The one or more non-transitory computer-readable storage media of claim 1, further comprising: responsive to determining that the first set of values are the same, performing, by the at least one processor, a second iteration that includes computing the value based on a second set of MSBs for each of the plurality of data sets, the second set of MSBs being larger than the first set of MSBs.
Claim 3[a]	The one or more non-transitory computer-readable storage media of claim 2, further storing instructions, which, when executed by the at least one processor, cause the at least one processor to perform operations comprising: examining a second set of values computed for the plurality of data sets in the second iteration to determine whether the maximum value is present among the second set of values; and

Claim Designation	Claim Language
Claim 3[b]	responsive to identifying the maximum value among the second set of values, performing, by the at least one processor, the full precision computation of the value for a data set, of the plurality of data sets, that exhibited the maximum value in the second iteration.
Claim 4	The one or more non-transitory computer-readable storage media of claim 2, wherein the computing in each of the first iteration and the second iteration employs a convolution and a pooling.
Claim 5	The one or more non-transitory computer-readable storage media of claim 4, wherein the convolution is a N×N convolution, where N is any integer.
Claim 6	The one or more non-transitory computer-readable storage media of claim 4, wherein the pooling is a N×N pooling, where N is any integer.
Claim 7	The one or more non-transitory computer-readable storage media of claim 4, wherein the convolution is a 3×3 convolution, and the pooling is a 2×2 pooling.
Claim 8	The one or more non-transitory computer-readable storage media of claim 2, wherein at least one of the first iteration and the second iteration is performed with a precision less than that of the full precision computation.
Claim 9	The one or more non-transitory computer-readable storage media of claim 8, wherein the precision is 8-bit precision.
Claim 10	The one or more non-transitory computer-readable storage media of claim 1, wherein the CNN is employed to analyze an image.
Claim 11[a]	The one or more non-transitory computer-readable storage media of claim 1, wherein: the first iteration computes a value that approximates the full precision computation of the value; and
Claim 11[b]	the full precision computation is performed on the data set that includes less data than the plurality of data sets.

APPENDIX OF EXHIBITS

Exhibit 1001	U.S. Patent No. 11,775,831 ("the '831 Patent")
Exhibit 1002	File History of the '831 Patent
Exhibit 1003	Declaration of Eli Saber, Ph.D. ("*Dec*")
Exhibit 1004	Takayuki Ujiie *et al*., *Approximated Prediction Strategy for Reducing Power Consumption of Convolutional Neural Network Processor,* 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (2016) ("*Ujiie*")
Exhibit 1005	Bert Moons *et al*., *Energy-Efficient ConvNets Through Approximate Computing,* IEEE Winter Conference on Applications of Computer Vision (2016) ("*Moons*")
Exhibit 1006	Himanshu Kaul *et al*., *A 21.5M-Query-Vectors/s 3.37nJ/Vector Reconfigurable k-Nearest-Neighbor Accelerator with Adaptive Precision in 14nm Tri-Gate CMOS*, 2016 IEEE International Solid-State Circuits Conference (2016) ("*Kaul*")
Exhibit 1007	U.S. Patent Application Publication No. 2016/0259995 A1 to Ishii *et al*. ("*Ishii*")
Exhibit 1008	U.S. Patent Application Publication No. 2016/0259994 A1 to Ravindran *et al*. ("*Ravindran*")
Exhibit 1009	Declaration of June Munford ("*Munford Dec.*")
Exhibit 1010	Declaration of Gordon MacPherson ("*MacPherson Dec.*")
Exhibit 1011	Matthieu Courbariaux *et al*., *Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or-1*, arXiv: 1602.02830v3 [cs.LG] (March 17, 2016) ("*Courbariaux*")
Exhibit 1012	Minkyu Kim *et al*., *An Energy-Efficient Deep Convolutional Neural Network Accelerator Featuring Conditional Computing and Low External Memory Access,* IEEE Journal of Solid-State Circuits, Vol. 56, No. 3 (2020) ("*Kim*")
Exhibit 1013	Yann LeCun *et al*., *Gradient-Based Learning Applied to Document Recognition,* Proceedings of the IEEE (1998) ("*LeCun 1*")
Exhibit 1014	Jiuxiang Gu *et al*., *Recent Advances in Convolutional Neural Networks,* arXiv: 1512.07108v1 [cs.CV] (December 22, 2015) ("*Gu*")
Exhibit 1015	Dominik Scherer *et al*., *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition,* 20th International Conference on Artificial Neural Networks (ICANN), Thessaloniki, Greece (2010) ("*Scherer*")

Exhibit 1016	U.S. Patent No. 9,886,948 to Garimella *et al*. ("*Garimella*")
Exhibit 1017	Dingjun Yu *et al*., *Mixed Pooling for Convolutional Neural Networks,* The 9th International Conference on Rough Sets and Knowledge Technology (2014) ("*Yu*")
Exhibit 1018	Alex Krizhevsky *et al*., *ImageNet Classification with Deep Convolutional Neural Networks,* Advances in Neural Information Processing Systems (2012) ("*Krizhevsky*")
Exhibit 1019	Thomas L. Floyd, *Digital Fundamentals, 11th ed.,* Pearson Education Ltd. 2015, excerpts ("*Floyd*")
Exhibit 1020	*Intentionally left blank*
Exhibit 1021	Yoshua Bengio *et al*., *Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation*, arXiv: 1308.3432v1 [cs.LG] (August 15, 2013) ("*Bengio 1*")
Exhibit 1022	Emmanuel Bengio *et al*., *Conditional Computation in Neural Networks for Faster Models,* arXiv: 1511.06297v1 [cs.LG] (November 19, 2015) ("*Bengio 2*")
Exhibit 1023	Vincent Vanhoucke *et al*., *Improving the speed of neural networks on CPUs,* Deep Learning and Unsupervised Feature Learning Workshop, NIPS (2011) ("*Vanhoucke*")
Exhibit 1024	Patrick Judd *et al*., *Reduced-Precision Strategies for Bounded Memory in Deep Neural Nets,* International Conference on Learning Representations 2016 (2016) ("*Judd*")
Exhibit 1025	Microsoft Computer Dictionary, 3rd ed., 1997, excerpts ("*Computer Dictionary*")
Exhibit 1026	Song Han *et al*., *EIE: Efficient Inference Engine on Compressed Deep Neural Network*, arXiv:1602.01528v2 [cs.CV] (May 3, 2016) ("*Han*")
Exhibit 1027	Vinay K. Chippa *et al*., *Analysis and Characterization of Inherent Application Resilience for Approximate Computing*, DAC '13: Proceedings of the 50th Annual Design Automation Conference (2013) ("*Chippa*")
Exhibit 1028	Pakorn Watanachaturaporn *et al*., *Hyperspectral Image Classification Using Support Vector Machines: A Comparison with Decision Tree and Neural Network Classifiers*, ASPRS 2005 Annual Conference (2005) ("*Watanachaturaporn*")
Exhibit 1029	Wei Wen *et al*., *An EDA Framework for Large Scale Hybrid Neuromorphic Computing Systems*, DAC '15: Proceedings of the 52nd Annual Design Automation Conference (2015) ("*Wen*")
Exhibit 1030	*Intentionally left blank*

Exhibit 1031	Yann LeCun *et al*., *Deep learning,* Nature Vol. 521 (May 28, 2015) ("*LeCun 2*")
Exhibit 1032	Ahmed AK Tahir, *Integrating artificial neural network and classical methods for unsupervised classification of optical remote sensing data,* EURASIP Journal on Advances in Signal Processing (2012) ("*Tahir*")

CERTIFICATION OF WORD COUNT

The undersigned certifies pursuant to 37 C.F.R. §42.24 that the foregoing Petition for *Inter Partes* Review, excluding any table of contents, mandatory notices under 37 C.F.R. §42.8, certificates of service or word count, or appendix of exhibits, contains 9,958 words according to the word-processing program used to prepare this document (Microsoft Word).

Dated: September 26, 2025

By: /s/ Paul R. Hart
Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER

CERTIFICATE OF SERVICE ON PATENT OWNER
UNDER 37 C.F.R. § 42.105

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that on September 26, 2025, a complete and entire copy of this Petition for *Inter Partes* Review and Exhibits were provided via Federal Express to the Patent Owner by serving the correspondence address of record for the '831 Patent:

FISH & RICHARDSON PC
PO Box 1022
Minneapolis, MN 55440-1022

Further, a courtesy copy of this Petition for *Inter Partes* Review was sent via email to Patent Owner's litigation counsel:

Jay P. Kesan (jay.kesan@kkd-law.com)
Cecil E. Key (cecil.key@kkd-law.com)
Alfonso Garcia Chan (achan@cahill.com)
Britton F. Davis (bdavis@cahill.com)
Christopher C. Campbell (ccampbell@cahill.com)
Halima Shukri Ndai (hndai@cahill.com)

Respectfully submitted,

By: /s/ Paul R. Hart
Paul R. Hart, Reg. No. 59,646

COUNSEL FOR PETITIONER