



(51) International Patent Classification:
G06N 3/08 (2006.01) G06N 3/04 (2006.01)

(21) International Application Number:
PCT/US2018/020863

(22) International Filing Date:
05 March 2018 (05.03.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/472,841 17 March 2017 (17.03.2017) US
15/908,870 01 March 2018 (01.03.2018) US

(71) Applicant: NEC LABORATORIES AMERICA, INC.
[US/US]; 4 Independence Way, Suite 200, Princeton, New
Jersey 08540 (US).

(72) Inventors: CHOI, Wongun; 16 Willow Street, Lexington,
Massachusetts 02421 (US). CHANDRAKER, Manmo-
han; 444 Saratoga Avenue, Santa Clara, California 95050
(US). CHEN, Guobin; 1543 Burrell Court, San Jose, Cali-
fornia 95126 (US). YU, Xiang; 131 Concord Circle, Moun-
tain View, California 94040 (US).

(74) Agent: KOLODKA, Joseph; NEC Laboratories America,
Inc., 4 Independence Way, Suite 200, Princeton, New Jersey
08540 (US).

(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP,
KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,
MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,

(54) Title: LEARNING EFFICIENT OBJECT DETECTION MODELS WITH KNOWLEDGE DISTILLATION

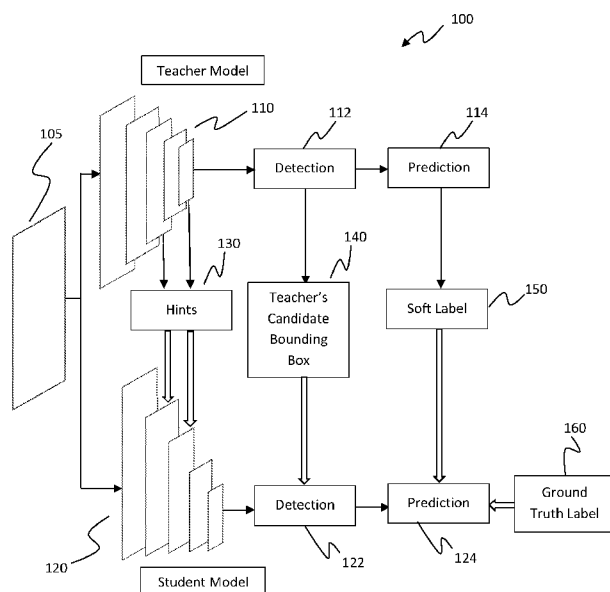


FIG. 1

(57) Abstract: A computer-implemented method executed by at least one processor for training fast models for real-time object detection with knowledge transfer is presented. The method includes employing a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection, inputting a plurality of images into the Faster R-CNN, and training the Faster R-CNN by learning a student model from a teacher model by employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes, employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model, and employing a confidence-weighted binary activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,
SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN,
TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

LEARNING EFFICIENT OBJECT DETECTION MODELS WITH KNOWLEDGE DISTILLATION

RELATED APPLICATION INFORMATION

[0001] This application claims priority to Provisional Application No. 62/472,841, filed on March 17, 2017 and U.S. Utility Application Serial No. 15/908,870 filed March 1 2018, incorporated herein by reference in its entirety.

BACKGROUND

Technical Field

[0002] The present invention relates to neural networks and, more particularly, to learning efficient object detection models with knowledge distillation in neural networks.

Description of the Related Art

[0003] Recently there has been a tremendous increase in the accuracy of object detection by employing deep convolutional neural networks (CNNs). This has made visual object detection an attractive possibility for domains ranging from surveillance to autonomous driving. However, speed is a key requirement in many applications, which fundamentally contends with demands on accuracy. Thus, while advances in object detection have relied on increasingly deeper architectures, such architectures are associated with an increase in computational expense at runtime.

SUMMARY

[0004] A computer-implemented method executed by at least one processor for training fast models for real-time object detection with knowledge transfer is presented. The method includes employing a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection, inputting a plurality of images into the Faster R-CNN, and training the Faster R-CNN by learning a student model from a teacher model by employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes, employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model, and employing a confidence-weighted binary activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

[0005] A system for training fast models for real-time object detection with knowledge transfer is also presented. The system includes a memory and a processor in communication with the memory, wherein the processor is configured to employ a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection, input a plurality of images into the Faster R-CNN, and train the Faster R-CNN by learning a student model from a teacher model by: employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes, employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model, and employing a confidence-weighted binary

activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

[0006] A non-transitory computer-readable storage medium comprising a computer-readable program is presented for training fast models for real-time object detection with knowledge transfer, wherein the computer-readable program when executed on a computer causes the computer to perform the steps of employing a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection, inputting a plurality of images into the Faster R-CNN, and training the Faster R-CNN by learning a student model from a teacher model by employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes, employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model, and employing a confidence-weighted binary activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

[0007] These and other features and advantages will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF DRAWINGS

[0008] The disclosure will provide details in the following description of preferred embodiments with reference to the following figures wherein:

[0009] FIG. 1 is a block/flow diagram illustrating a knowledge distillation structure, in accordance with embodiments of the present invention;

[0010] FIG. 2 is a block/flow diagram illustrating a real-time object detection framework, in accordance with embodiments of the present invention;

[0011] FIG. 3 is a block/flow diagram illustrating a Faster Region-based convolutional neural network (R-CNN), in accordance with embodiments of the present invention;

[0012] FIG. 4 is a block/flow diagram illustrating a method for training fast models for real-time object detection with knowledge transfer, in accordance with embodiments of the present invention;

[0013] FIG. 5 is an exemplary processing system for training fast models for real-time object detection with knowledge transfer, in accordance with embodiments of the present invention;

[0014] FIG. 6 is a block/flow diagram of a method for training fast models for real-time object detection with knowledge transfer in Internet of Things (IoT) systems/devices/infrastructure, in accordance with embodiments of the present invention; and

[0015] FIG. 7 is a block/flow diagram of exemplary IoT sensors used to collect data/information related to training fast models for real-time object detection with knowledge transfer, in accordance with embodiments of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0016] In the exemplary embodiments of the present invention, methods and devices for implementing deep neural networks are introduced. Deep neural networks have

recently exhibited state-of-the-art performance in computer vision tasks such as image classification and object detection. Moreover, recent knowledge distillation approaches are aimed at obtaining small and fast-to-execute models, and such approaches have shown that a student network could imitate a soft output of a larger teacher network or ensemble of networks. Thus, knowledge distillation approaches have been incorporated into neural networks.

[0017] While deeper networks are easier to train, tasks such as object detection for a few categories might not necessarily need such model capacity. As a result, several conventional techniques in image classification employ model compression, where weights in each layer are decomposed, followed by layer-wise reconstruction or fine-tuning to recover some of the accuracy. This results in significant speed-ups, but there is often a gap between the accuracies of original and compressed models, which can be large when using compressed models for more complex problems such as object detection. On the other hand, knowledge distillation techniques illustrate that a shallow or compressed model trained to mimic a behavior of a deeper or more complex model can recover some or all of the accuracy drop.

[0018] In the exemplary embodiments of the present invention, a method for training fast models for object detection with knowledge transfer is introduced. First, a weighted cross entropy loss layer is employed for classification that accounts for an imbalance in the impact of misclassification for background classes as opposed to between object classes. Next, a prediction vector of a bounding box regression of a teacher model is employed as a target for a student model, through an L2 boundary loss. Further, under-fitting is addressed by employing a binary activation loss layer for intermediate layers

that allows gradients that account for the relative confidence of teacher and student models. Moreover, adaptation layers can be employed for domain specific fitting that allows student models to learn from distribution of neurons in the teacher model.

[0019] FIG. 1 is a block/flow diagram 100 illustrating a knowledge distillation structure, in accordance with embodiments of the present invention.

[0020] A plurality of images 105 are input into the teacher model 110 and the student model 120. Hint learning module 130 can be employed to aid the student model 120. The teacher model 110 interacts with a detection module 112 and a prediction module 114, and the student model 120 interacts with a detection module 122 and a prediction module 124. Bounding box regression module 140 can also be used to adjust a location and a size of the bounding box. The prediction modules 114, 116 communicate with soft label module 150 and ground truth module 160.

[0021] The teacher model 110 and the student model 120 are models that are trained to output a predetermined output with respect to a predetermined input, and may include, for example, neural networks. A neural network refers to a recognition model that simulates a computation capability of a biological system using a large number of artificial neurons being connected to each other through edges. It is understood, however, that the teacher model 110 and student model 120 are not limited to neural networks, and may also be implemented in other types of networks and apparatuses.

[0022] The neural network uses artificial neurons configured by simplifying functions of biological neurons, and the artificial neurons may be connected to each other through edges having connection weights. The connection weights, parameters of the neural network, are predetermined values of the edges, and may also be referred to as

connection strengths. The neural network may perform a cognitive function or a learning process of a human brain through the artificial neurons. The artificial neurons may also be referred to as nodes.

[0023] A neural network may include a plurality of layers. For example, the neural network may include an input layer, a hidden layer, and an output layer. The input layer may receive an input to be used to perform training and transmit the input to the hidden layer, and the output layer may generate an output of the neural network based on signals received from nodes of the hidden layer. The hidden layer may be disposed between the input layer and the output layer. The hidden layer may change training data received from the input layer to an easily predictable value. Nodes included in the input layer and the hidden layer may be connected to each other through edges having connection weights, and nodes included in the hidden layer and the output layer may also be connected to each other through edges having connection weights. The input layer, the hidden layer, and the output layer may respectively include a plurality of nodes.

[0024] The neural network may include a plurality of hidden layers. A neural network including the plurality of hidden layers may be referred to as a deep neural network. Training the deep neural network may be referred to as deep learning. Nodes included in the hidden layers may be referred to as hidden nodes. The number of hidden layers provided in a deep neural network is not limited to any particular number.

[0025] The neural network may be trained through supervised learning. Supervised learning refers to a method of providing input data and output data corresponding thereto to a neural network, and updating connection weights of edges so that the output data corresponding to the input data may be output. For example, a model training apparatus

may update connection weights of edges among artificial neurons through a delta rule and error back-propagation learning.

[0026] Error back-propagation learning refers to a method of estimating a loss with respect to input data provided through forward computation, and updating connection weights to reduce a loss in a process of propagating the estimated loss in a backward direction from an output layer toward a hidden layer and an input layer. Processing of the neural network may be performed in an order of the input layer, the hidden layer, and the output layer. However, in the error back-propagation learning, the connection weights may be updated in an order of the output layer, the hidden layer, and the input layer. Hereinafter, according to an exemplary embodiment, training a neural network refers to training parameters of the neural network. Further, a trained neural network refers to a neural network to which the trained parameters are applied.

[0027] The teacher model 110 and the student model 120 may be neural networks of different sizes which are configured to recognize the same target. It is understood, however, that the teacher model 110 and the student model 120 are not required to be different sizes.

[0028] The teacher model 110 is a model that recognizes target data with a relatively high accuracy based on a sufficiently large number of features extracted from target data to be recognized. The teacher model 110 may be a neural network of a greater size than the student model 120. For example, the teacher model 110 may include a larger number of hidden layers, a larger number of nodes, or a combination thereof, compared to the student model 120.

[0029] The student model 120 may be a neural network of a smaller size than the teacher model 110. Due to the relatively small size, the student model 120 may have a higher recognition rate than the teacher model 110. The student model 120 may be trained using the teacher model 110 to provide output data of the teacher model 110 with respect to input data. For example, the output data of the teacher model 110 may be a value of logic output from the teacher model 110, a probability value, or an output value of a classifier layer derived from a hidden layer of the teacher model 110. Accordingly, the student model 120 having a higher recognition rate than the teacher model 110 while outputting the same value as that output from the teacher model 110 may be obtained. The foregoing process may be referred to as model compression. Model compression is a scheme of training the student model 120 based on output data of the teacher model 110, instead of training the student model 120 based on correct answer data corresponding to a true label.

[0030] A plurality of teacher models 110 may be used to train the student model 120. At least one teacher model may be selected from the plurality of teacher models 110 and the student model 120 may be trained using the selected at least one teacher model. A process of selecting at least one teacher model from the plurality of teacher models 110 and training the student model 120 may be performed iteratively until the student model 120 satisfies a predetermined condition. In this example, at least one teacher model selected to be used to train the student model 120 may be newly selected each time a training process is performed. For example, one or more teacher models may be selected to be used to train the student model 120.

[0031] Additionally, each item in a batch can be classified by obtaining its feature set and then executing each classifier in a set of existing classifiers on such feature set, thereby producing corresponding classification predictions. Such predictions are intended to predict the ground truth label 160 that would be identified for the corresponding item if the item were to be classified manually. In the present embodiments, the “ground truth label” 160 (sometimes referred to herein simply as the label) represents a specific category (hard label) into which the specific item should be placed. Depending upon the particular embodiment, the classification predictions either identify particular categories to which the corresponding item should be assigned (sometimes referred to as hard classification predictions) or else constitute classification scores which indicate how closely related the items are to particular categories (sometimes referred to as soft classification predictions). Such a soft classification prediction preferably represents the probability that the corresponding item belongs to a particular category. It is noted that either hard or soft classification predictions can be generated irrespective of whether the ground truth labels are hard labels or soft labels, although often the predictions and labels will be of the same type.

[0032] In one exemplary embodiment, a classification approach can be used to train a classifier on known emotional responses. The video or image sequences of one or more subjects exhibiting an emotion or behavior are labeled based on ground truth labeling 160. These labels are automatically generated for video sequences capturing a subject after the calibration task is used to trigger an emotion. Using the classification technique, the response time, difficulty level of the calibration task, and the quality of the response to the task can be used as soft-labels 150 for indicating the emotion. The ground truth data

is used in a learning stage that trains the classifier for detecting future instances of such behaviors (detection stage). Features and metrics can be extracted from the subjects during both the learning and detection stages.

[0033] FIG. 2 is a block/flow diagram 200 illustrating a real-time object detection framework, in accordance with embodiments of the present invention.

[0034] The diagram 200 includes a plurality of images 105 input into the region proposal network 210 and the region classification network 220. Processing involving soft labels 150 and ground truth labels 160 can aid the region proposal network 210 and the region classification network 220 in obtaining desired results 250.

[0035] FIG. 3 is a block/flow diagram illustrating a Faster Region-based convolutional neural network (R-CNN), in accordance with embodiments of the present invention.

[0036] In the exemplary embodiments of the present invention, the Faster R-CNN can be adopted as the object detection framework. Faster R-CNN can include three modules, that is, a feature extractor 310, a proposal or candidate generator 320, and a box classifier 330. The feature extractor 310 allows for shared feature extraction through convolutional layers. The proposal generator 320 can be, e.g., a region proposal network (RPN) 210 that generates object proposals. The proposal generator 320 can include an object classification module 322 and a module 324 that is to keep or reject the proposal. The box classifier 330 can be, e.g., a classification and regression network (RCN) 220 that returns a detection score of the region. The box classifier 330 can include a multiway classification module 332 and a box regression module 334.

[0037] In order to achieve highly accurate object detection results 250, it is necessary to learn strong models for all the three components 310, 320, 330. Strong but efficient

student object detectors are learned by using the knowledge of a high capacity teacher detection networks in all the three components 310, 320, 330.

[0038] Firstly, hint based learning can be employed that encourages a feature representation of a student network/model that is similar to that of the teacher network/model. A new loss function, e.g., a Binary Activation Loss function or layer, is employed that is more stable than L2 and puts more weight on activated neurons. Secondly, stronger classification modules are learned in both RPN 210 and RCN 220 by using the knowledge distillation framework of FIG. 1. In order to handle category imbalance issues in object detection, a weighted cross entropy loss layer is applied for the distillation framework of FIG. 1. Finally, the teacher's regression output is transferred as a form of upper bound, e.g., if the student's regression output is better than that of teacher, no loss is applied.

[0039] The overall learning objective can be written as follow:

$$\begin{aligned} L_{RPN} &= L_{RPN}^{CLS} + L_{hard}^{REG} + \lambda_2 L_{loc}^{REG} \\ L_{RCN} &= L_{RPN}^{CLS} + L_{hard}^{REG} + \lambda_2 L_{loc}^{REG} \\ L &= L_{RPN} + L_{RCN} + L_{Hint} \end{aligned} \quad (1)$$

[0040] Where L_{RPN}^{CLS} , L_{RCN}^{CLS} denotes the loss function defined in Eq. 2,

L_{Hint} denotes the loss function defined in Eq. 5, L_{loc}^{REG} is defined in Eq. 4 and

L_{hard}^{REG} is the Smooth L1.

[0041] Knowledge distillation classification is introduced for training a classification network by using predictions of the teacher networks to guide the training of the student

model. Assume the following dataset $\{x_i, y_i\}, i=1,2,\dots,n$ where $x_i \in \mathcal{X}$ is the input image and $y_i \in \mathcal{Y}$ is the label of the image.

[0042] Let t be the teacher model, $P_t = \text{softmax}(\frac{Z}{T})$ represent a prediction of the teacher model, while Z is an output of the last layer in t .

[0043] Similarly for the student network, assume: $P_s = \text{softmax}(\frac{V}{T})$.

[0044] The student network is trained to optimize the following loss function:

$$L_{CLS} = \lambda L_{hard}(P_s, y) + (1 - \lambda) L_{soft}(P_s, P_t) \quad (2)$$

[0045] where λ is the parameter to balance the hard loss and soft loss.

[0046] In conventional frameworks, both losses are cross entropy losses. Since P_t might be very close to the hard label, i.e., most of the probabilities are very close to 0. Conventional frameworks further introduced temperature parameter T to soften an output of the networks, which forces the production of a probability vector with relatively large values for each class. By learning from the soft label 150, the student network 120 could determine how the teacher network 110 tends to generalize and learn the relationship between different classes.

[0047] However, the process is different for the detection task. Although conventional works have proven that using L2 loss to match the logits before softmax is only a special case of distillation in a high temperature case, other conventional works have reported that L2 loss works better than the softened cross entropy loss for detection. The same phenomenon can be seen in experiments conducted employing the exemplary embodiments of the present invention. One cause of this is a difference between image

classification and object detection. In image classification, the only error is misclassification, e.g., misclassify “cat” in an image as a “dog.” However, in object detection, failing to distinguish background/foreground and inaccurate localization dominate the error, while a proportion of misclassification between different classes is not very large. On one hand, the soft labels 150 are still useful for object detection since they contain richer information about the extent of being a background/foreground. On the other hand, soft labels 150 can be quite noisy at high temperatures since they may provide misleading information of being another object.

[0048] To address this, the following class-weighted cross entropy loss is employed:

$$L_{soft}(P_s, P_t) = - \sum w_c P_t \log P_s \quad (3)$$

[0049] where Eq. (3) can use a larger weight for the background class and a relatively small weight for the other classes.

[0050] Regarding bounding box regression, apart from the classification layer, Faster R-CNN also employs bounding-box regression to adjust a location and a size of an input bounding box. The label of the bounding-box regression is the offsets of the input bounding-box and the ground truth. Learning from the teacher’s prediction may not be reasonable since it does not contain information from other classes or backgrounds. A good way to make use of the teacher’s prediction is to use it as the boundary of the student network. The prediction vector of bounding-box regression should be as close to the label as possible, or at least should be closer than the teacher’s prediction.

[0051] By following this technique, L2 with boundary loss to transfer knowledge is given as:

$$L_{REG} = \mathbb{1}(\|R_s - y_{loc}\|_2^2 + \alpha > \|R_t - y_{loc}\|_2^2) \quad (4)$$

$$(\|l_s - y_{loc}\|_2^2 + \|l_s - l_t\|_2^2)$$

[0052] where $\mathbb{1}(\cdot)$ is the indicator function, α is the margin parameter, y_{loc} denotes the regression label, R_s is the prediction of student network for regression task, R_t is the prediction of teacher network 110.

[0053] Therefore, the network is penalized only when the error of the student network 120 is larger than that of the teacher network 110.

[0054] Regarding hint learning, distillation only transfers knowledge from the last layer. In conventional works, it has been indicated that employing the intermediate representation of the teacher model 110 as hints can improve the training process and final performance of the student model 120.

[0055] Such works use L2 distance between feature vector V and Z ,

$$L_{Hint}(V, Z) = \|V - Z\|_2^2, \text{ to mimic the response of teacher model 110.}$$

[0056] The L2 loss takes each logits equally even from negative logits which will not be activated. When the teacher model 110 is more confident than the student model 120, a positive gradient should be passed to previous layers, otherwise a negative gradient is passed to previous layers.

[0057] Following this principle, the exemplary embodiments employ a Binary Activation loss, which learns according to the confidence of logit:

$$L_{Hint} = \mathbb{1}(Z_i < 0) \cdot V_i \cdot \left(\frac{1 + \text{sgn}(V_i)}{2} \right) + \mathbb{1}(Z_i \geq 0) \mathbb{1}(V_i \geq Z_i) (V_i - Z_i) \quad (5)$$

[0058] where $\mathbb{1}(\cdot)$ is the indicator function, $\text{sgn}(\cdot)$ is the sign function, V_i is one neuron in student's network and Z_i is one neuron in teacher's network 110.

[0059] Note that the input for Binary Activation loss should be before the rectified linear unit (ReLU) layer.

[0060] Distillation tends to solve the problem of generalization, in other words, the over-fitting problem. However, for shallower networks, the networks can also face an "under-fitting" problem. It's not easy for a shallow network to find the local minimal. Nevertheless, learning from hints could help the student model 120 converge faster. An adaption layer to map from layer L_s in student network 120 to layer L_t in teacher network 110 can be employed, even if the number of neurons are the same. The adaption layers serve as domain specific fittings, which could help the student model 120 learn a distribution of neurons in the teacher model 110 instead of from a direct response of each neuron.

[0061] Finally, regarding teacher and student networks 110, 120, in the exemplary embodiments of the present invention, Faster R-CNN can be employed as the model for real-time object detection. The detection includes shared convolutional layers, a Region Proposal Network (RPN) and a Region Classification Network (RCN). Each network includes a classification task and a regression task. Moreover, in the application of object detection, several important cues under the knowledge distillation framework are introduced to simplify the network structures and preserve the performance of the

networks. A new objective loss layer for the output feature to better match the source feature space is introduced for the knowledge distillation. Further, the adaptive domain transfer layer is introduced to regularize both the final output and intermediate layers of the student models 120. Thus, knowledge distillation and hint learning can be employed to generate the object detection area.

[0062] FIG. 4 is a block/flow diagram illustrating a method for training fast models for real-time object detection with knowledge transfer, in accordance with embodiments of the present invention.

[0063] At block 401, a Faster Region-based Convolutional Neural Network (R-CNN) is employed as an objection detection framework for performing the real-time object detection.

[0064] At block 403, a plurality of images are input into the Faster R-CNN.

[0065] At block 405, the Faster R-CNN is trained by learning a student model from a teacher model by blocks 407, 409, 411.

[0066] At block 407, a weighted cross-entropy loss layer is employed for classification accounting for an imbalance between background classes and object classes.

[0067] At block 409, a boundary loss layer is employed to enable transfer of knowledge of bounding box regression from the teacher model to the student model.

[0068] At block 411, a confidence-weighted binary activation loss layer is employed to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

[0069] Networks can represent all sorts of systems in the real world. For example, the Internet can be described as a network where the nodes are computers or other devices

and the edges are physical (or wireless, even) connections between the devices. The World Wide Web is a huge network where the pages are nodes and links are the edges. Other examples include social networks of acquaintances or other types of interactions, networks of publications linked by citations, transportation networks, metabolic networks, communication networks, and Internet of Things (IoT) networks. The exemplary embodiments of the present invention can refer to any such networks without limitation.

[0070] In summary, the exemplary embodiments of the present invention solve the problem of achieving object detection at an accuracy comparable to complex deep learning models, while maintaining speeds similar to a simpler deep learning model. The exemplary embodiments of the present invention also address the problem of achieving object detection accuracy comparable to high resolution images, while retaining the speed of a network that accepts low resolution images. The exemplary embodiments of the present invention introduce a framework for distillation in deep learning for complex object detection tasks that can transfer knowledge from a network with a large number of parameters to a compressed one. A weighted cross-entropy loss layer is employed that accounts for imbalance between background and other object classes. An L2 boundary loss layer is further employed to achieve distillation for bounding box regression. Also, a binary activation loss layer is employed to address the problem of under-fitting.

[0071] Moreover, the advantages of the exemplary embodiments are at least as follows: the exemplary embodiments retain accuracy similar to a complex model, while achieving speeds similar to a compressed model, the exemplary embodiments can achieve accuracy similar to high resolution images while working with low resolution

images, resulting in a significant speedup, and the exemplary embodiments can transfer knowledge from a deep model to a shallower one, allowing for faster speeds at the same training effort. Further advantages of the exemplary embodiments include the ability to design an effective framework that can transfer knowledge from a more expensive model to a cheaper one, allowing faster speed with minimal loss in accuracy, the ability to learn from low resolution images by mimicking the behavior of a model trained on high resolution images, allowing high accuracy at lower computational cost, taking into consideration imbalances between classes in detection that allows for accuracy improvement by weighing the importance of the background class, bounding box regression that allows transferring knowledge of better localization accuracy, and better training of intermediate layers through confidence-weighted binary activation loss that allows for higher accuracy.

[0072] Therefore, the framework allows for transferring knowledge from a more complex deep model to a less complex one. This framework is introduced for the complex task of object detection, by employing a novel weighted cross-entropy loss layer to balance the effects of background and other object classes, an L2 boundary loss layer to transfer the knowledge of bounding box regression from the teacher model to the student model, and a confidence-weighted binary activation loss to more effectively train the intermediate layers of the student model to achieve similar distribution of neurons as the teacher model.

[0073] FIG. 5 is an exemplary processing system for training fast models for real-time object detection with knowledge transfer, in accordance with embodiments of the present invention.

[0074] The processing system includes at least one processor (CPU) 504 operatively coupled to other components via a system bus 502. A cache 506, a Read Only Memory (ROM) 508, a Random Access Memory (RAM) 510, an input/output (I/O) adapter 520, a network adapter 530, a user interface adapter 540, and a display adapter 550, are operatively coupled to the system bus 502. Additionally, a Faster R-CNN network 501 for employing object detection is operatively coupled to the system bus 502. The Faster R-CNN 501 achieves object detection by employing a weighted cross-entropy loss layer 601, an L2 boundary loss layer 603, and a confidence-weighted binary activation loss layer 605.

[0075] A storage device 522 is operatively coupled to system bus 502 by the I/O adapter 520. The storage device 522 can be any of a disk storage device (e.g., a magnetic or optical disk storage device), a solid state magnetic device, and so forth.

[0076] A transceiver 532 is operatively coupled to system bus 502 by network adapter 530.

[0077] User input devices 542 are operatively coupled to system bus 502 by user interface adapter 540. The user input devices 542 can be any of a keyboard, a mouse, a keypad, an image capture device, a motion sensing device, a microphone, a device incorporating the functionality of at least two of the preceding devices, and so forth. Of course, other types of input devices can also be used, while maintaining the spirit of the present invention. The user input devices 542 can be the same type of user input device or different types of user input devices. The user input devices 542 are used to input and output information to and from the processing system.

[0078] A display device 552 is operatively coupled to system bus 502 by display adapter 550.

[0079] Of course, the Faster R-CNN network processing system may also include other elements (not shown), as readily contemplated by one of skill in the art, as well as omit certain elements. For example, various other input devices and/or output devices can be included in the system, depending upon the particular implementation of the same, as readily understood by one of ordinary skill in the art. For example, various types of wireless and/or wired input and/or output devices can be used. Moreover, additional processors, controllers, memories, and so forth, in various configurations can also be utilized as readily appreciated by one of ordinary skill in the art. These and other variations of the Faster R-CNN network processing system are readily contemplated by one of ordinary skill in the art given the teachings of the present invention provided herein.

[0080] FIG. 6 is a block/flow diagram of a method for training fast models for real-time object detection with knowledge transfer in Internet of Things (IoT) systems/devices/infrastructure, in accordance with embodiments of the present invention.

[0081] According to some embodiments of the invention, an advanced neural network is implemented using an IoT methodology, in which a large number of ordinary items are utilized as the vast infrastructure of a neural network.

[0082] IoT enables advanced connectivity of computing and embedded devices through internet infrastructure. IoT involves machine-to-machine communications (M2M), where it is important to continuously monitor connected machines to detect any anomaly or bug, and resolve them quickly to minimize downtime.

[0083] The neural network 501 can be incorporated, e.g., into wearable, implantable, or ingestible electronic devices and Internet of Things (IoT) sensors. The wearable, implantable, or ingestible devices can include at least health and wellness monitoring devices, as well as fitness devices. The wearable, implantable, or ingestible devices can further include at least implantable devices, smart watches, head-mounted devices, security and prevention devices, and gaming and lifestyle devices. The IoT sensors can be incorporated into at least home automation applications, automotive applications, user interface applications, lifestyle and/or entertainment applications, city and/or infrastructure applications, toys, healthcare, fitness, retail tags and/or trackers, platforms and components, etc. The neural network 501 described herein can be incorporated into any type of electronic devices for any type of use or application or operation.

[0084] IoT (Internet of Things) is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.

[0085] IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment. IoT systems enhance data collection, automation, operations, and much more through smart devices and powerful enabling technology.

[0086] IoT systems allow users to achieve deeper automation, analysis, and integration within a system. IoT improves the reach of these areas and their accuracy. IoT utilizes existing and emerging technology for sensing, networking, and robotics. Features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device

use. In various embodiments, the neural network 501 of the present invention can be incorporated into a variety of different devices and/or systems. For example, the neural network 501 can be incorporated into wearable or portable electronic devices 830. Wearable/portable electronic devices 830 can include implantable devices 831, such as smart clothing 832. Wearable/portable devices 830 can include smart watches 833, as well as smart jewelry 834. Wearable/portable devices 830 can further include fitness monitoring devices 835, health and wellness monitoring devices 837, head-mounted devices 839 (e.g., smart glasses 840), security and prevention systems 841, gaming and lifestyle devices 843, smart phones/tablets 845, media players 847, and/or computers/computing devices 849.

[0087] The neural network 501 of the present invention can be further incorporated into Internet of Thing (IoT) sensors 810 for various applications, such as home automation 821, automotive 823, user interface 825, lifestyle and/or entertainment 827, city and/or infrastructure 829, retail 811, tags and/or trackers 813, platform and components 815, toys 817, and/or healthcare 819. The IoT sensors 810 can communicate with the neural network 501. Of course, one skilled in the art can contemplate incorporating such neural network 501 formed therein into any type of electronic devices for any types of applications, not limited to the ones described herein.

[0088] FIG. 7 is a block/flow diagram of exemplary IoT sensors used to collect data/information related to training fast models for real-time object detection with knowledge transfer, in accordance with embodiments of the present invention.

[0089] IoT loses its distinction without sensors. IoT sensors act as defining instruments which transform IoT from a standard passive network of devices into an active system capable of real-world integration.

[0090] The IoT sensors 810 can be connected via neural network 501 to transmit information/data, continuously and in real-time, to any type of neural network 501. Exemplary IoT sensors 810 can include, but are not limited to, position/presence/proximity sensors 901, motion/velocity sensors 903, displacement sensors 905, such as acceleration/tilt sensors 906, temperature sensors 907, humidity/moisture sensors 909, as well as flow sensors 910, acoustic/sound/vibration sensors 911, chemical/gas sensors 913, force/load/torque/strain/pressure sensors 915, and/or electric/magnetic sensors 917. One skilled in the art can contemplate using any combination of such sensors to collect data/information and input into the layers 601, 603, 605 of the neural network 501 for further processing. One skilled in the art can contemplate using other types of IoT sensors, such as, but not limited to, magnetometers, gyroscopes, image sensors, light sensors, radio frequency identification (RFID) sensors, and/or micro flow sensors. IoT sensors can also include energy modules, power management modules, RF modules, and sensing modules. RF modules manage communications through their signal processing, WiFi, ZigBee®, Bluetooth®, radio transceiver, duplexer, etc.

[0091] Moreover data collection software can be used to manage sensing, measurements, light data filtering, light data security, and aggregation of data. Data collection software uses certain protocols to aid IoT sensors in connecting with real-time, machine-to-machine networks. Then the data collection software collects data from

multiple devices and distributes it in accordance with settings. Data collection software also works in reverse by distributing data over devices. The system can eventually transmit all collected data to, e.g., a central server.

[0092] As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “circuit,” “module” or “system.” Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[0093] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical data storage device, a magnetic data storage device, or any suitable combination

of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can include, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0094] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

[0095] Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[0096] Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the “C” programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a

wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0097] Aspects of the present invention are described below with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the present invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks or modules.

[0098] These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks or modules.

[0099] The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the

computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks or modules.

[00100] It is to be appreciated that the term “processor” as used herein is intended to include any processing device, such as, for example, one that includes a CPU (central processing unit) and/or other processing circuitry. It is also to be understood that the term “processor” may refer to more than one processing device and that various elements associated with a processing device may be shared by other processing devices.

[00101] The term “memory” as used herein is intended to include memory associated with a processor or CPU, such as, for example, RAM, ROM, a fixed memory device (e.g., hard drive), a removable memory device (e.g., diskette), flash memory, etc. Such memory may be considered a computer readable storage medium.

[00102] In addition, the phrase “input/output devices” or “I/O devices” as used herein is intended to include, for example, one or more input devices (e.g., keyboard, mouse, scanner, etc.) for entering data to the processing unit, and/or one or more output devices (e.g., speaker, display, printer, etc.) for presenting results associated with the processing unit.

[00103] The foregoing is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the patent laws. It is to be understood that the embodiments shown and described herein are only illustrative of the principles of the present invention and that those skilled in the art may implement various modifications without departing from the scope and spirit of the invention. Those skilled in the art could

implement various other feature combinations without departing from the scope and spirit of the invention. Having thus described aspects of the invention, with the details and particularity required by the patent laws, what is claimed and desired protected by Letters Patent is set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A computer-implemented method executed by at least one processor for training fast models for real-time object detection with knowledge transfer, the method comprising:

employing a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection;

inputting a plurality of images into the Faster R-CNN; and

training the Faster R-CNN by learning a student model from a teacher model by:

employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes;

employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model; and

employing a confidence-weighted binary activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

2. The method of claim 1, further comprising adopting hint-based learning that enables a feature representation of the student model to be similar to a feature representation of the teacher model.

3. The method of claim 2, further comprising enabling the hint-based learning to provide hints to the student model for finding local minima.

4. The method of claim 1, further comprising applying a larger weight for the background classes and a smaller weight for the object classes in the weighted cross-entropy loss layer.

5. The method of claim 1, further comprising setting a prediction vector of the bounding box regression to approximate a class label in the boundary loss layer.

6. The method of claim 1, further comprising allowing the student model to learn from a bounding box location of the teacher model in the boundary loss layer.

7. The method of claim 1, further comprising applying a positive gradient to the intermediate layers of the student model when a confidence of the teacher model is greater than a confidence of the student model in the confidence-weighted binary activation loss layer.

8. A system for training fast models for real-time object detection with knowledge transfer, the system comprising:

a memory; and

a processor in communication with the memory, wherein the processor runs program code to:

employ a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection;
input a plurality of images into the Faster R-CNN; and

train the Faster R-CNN by learning a student model from a teacher model by:

employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes;

employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model; and

employing a confidence-weighted binary activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

9. The system of claim 8, wherein hint-based learning is adopted that enables a feature representation of the student model to be similar to a feature representation of the teacher model.

10. The system of claim 9, wherein the hint-based learning is enabled to provide hints to the student model for finding local minima.

11. The system of claim 8, wherein a larger weight is applied for the background classes and a smaller weight is applied for the object classes in the weighted cross-entropy loss layer.

12. The system of claim 8, wherein a prediction vector of the bounding box regression is set to approximate a class label in the boundary loss layer.

13. The system of claim 8, wherein the student model is permitted to learn from a bounding box location of the teacher model in the boundary loss layer.

14. The system of claim 8, wherein a positive gradient is applied to the intermediate layers of the student model when a confidence of the teacher model is greater than a confidence of the student model in the confidence-weighted binary activation loss layer.

15. A non-transitory computer-readable storage medium comprising a computer-readable program for training fast models for real-time object detection with knowledge transfer, wherein the computer-readable program when executed on a computer causes the computer to perform the steps of:

employing a Faster Region-based Convolutional Neural Network (R-CNN) as an objection detection framework for performing the real-time object detection;

inputting a plurality of images into the Faster R-CNN; and

training the Faster R-CNN by learning a student model from a teacher model by:

employing a weighted cross-entropy loss layer for classification accounting for an imbalance between background classes and object classes;

employing a boundary loss layer to enable transfer of knowledge of bounding box regression from the teacher model to the student model; and

employing a confidence-weighted binary activation loss layer to train intermediate layers of the student model to achieve similar distribution of neurons as achieved by the teacher model.

16. The non-transitory computer-readable storage medium of claim 15, wherein hint-based learning is adopted that enables a feature representation of the student model to be similar to a feature representation of the teacher model.

17. The non-transitory computer-readable storage medium of claim 16, wherein the hint-based learning is enabled to provide hints to the student model for finding local minima.

18. The non-transitory computer-readable storage medium of claim 15, wherein a larger weight is applied for the background classes and a smaller weight is applied for the object classes in the weighted cross-entropy loss layer.

19. The non-transitory computer-readable storage medium of claim 15, wherein a prediction vector of the bounding box regression is set to approximate a class label in the boundary loss layer.

20. The non-transitory computer-readable storage medium of claim 15, wherein the student model is permitted to learn from a bounding box location of the teacher model in the boundary loss layer.

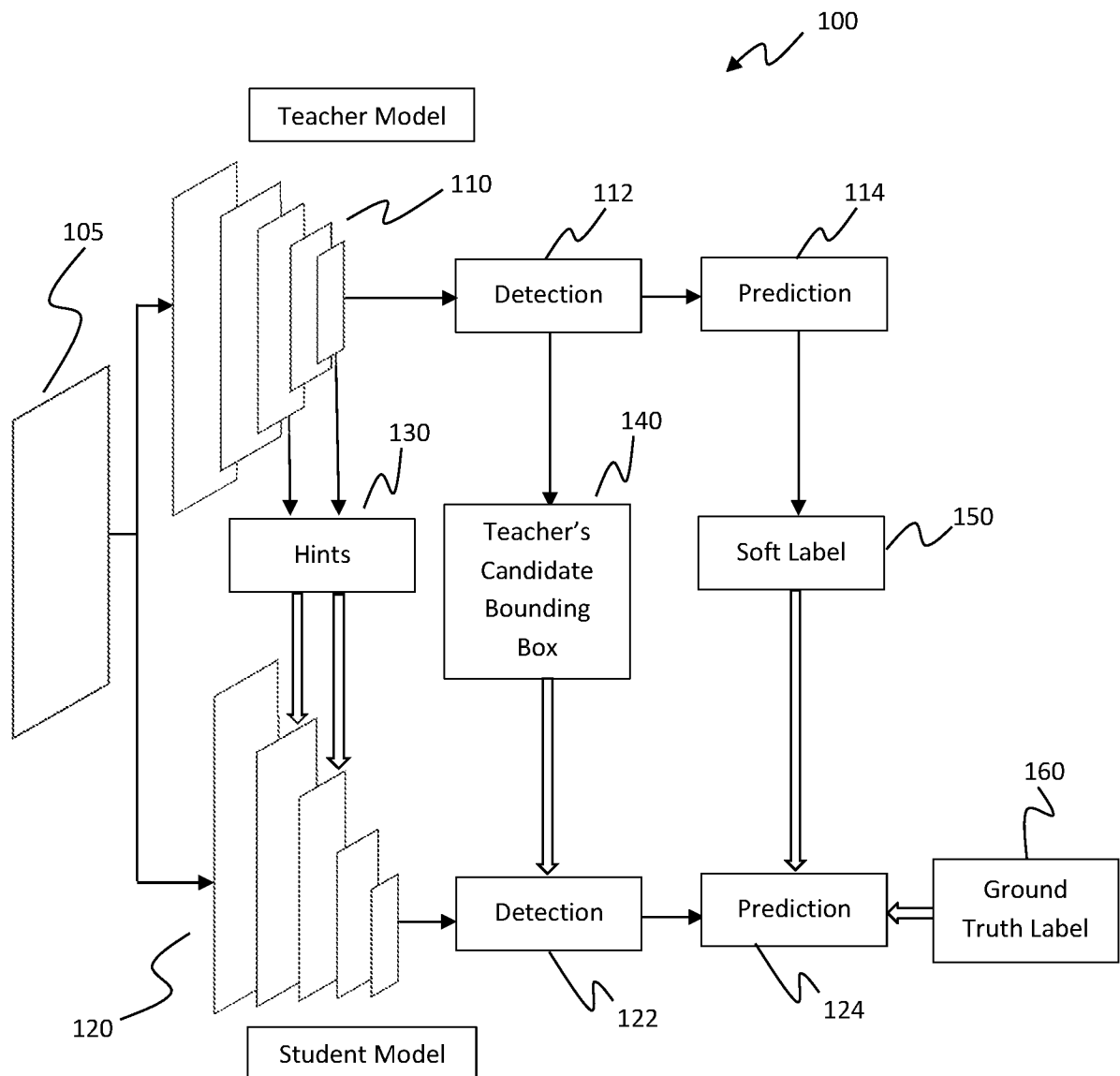


FIG. 1

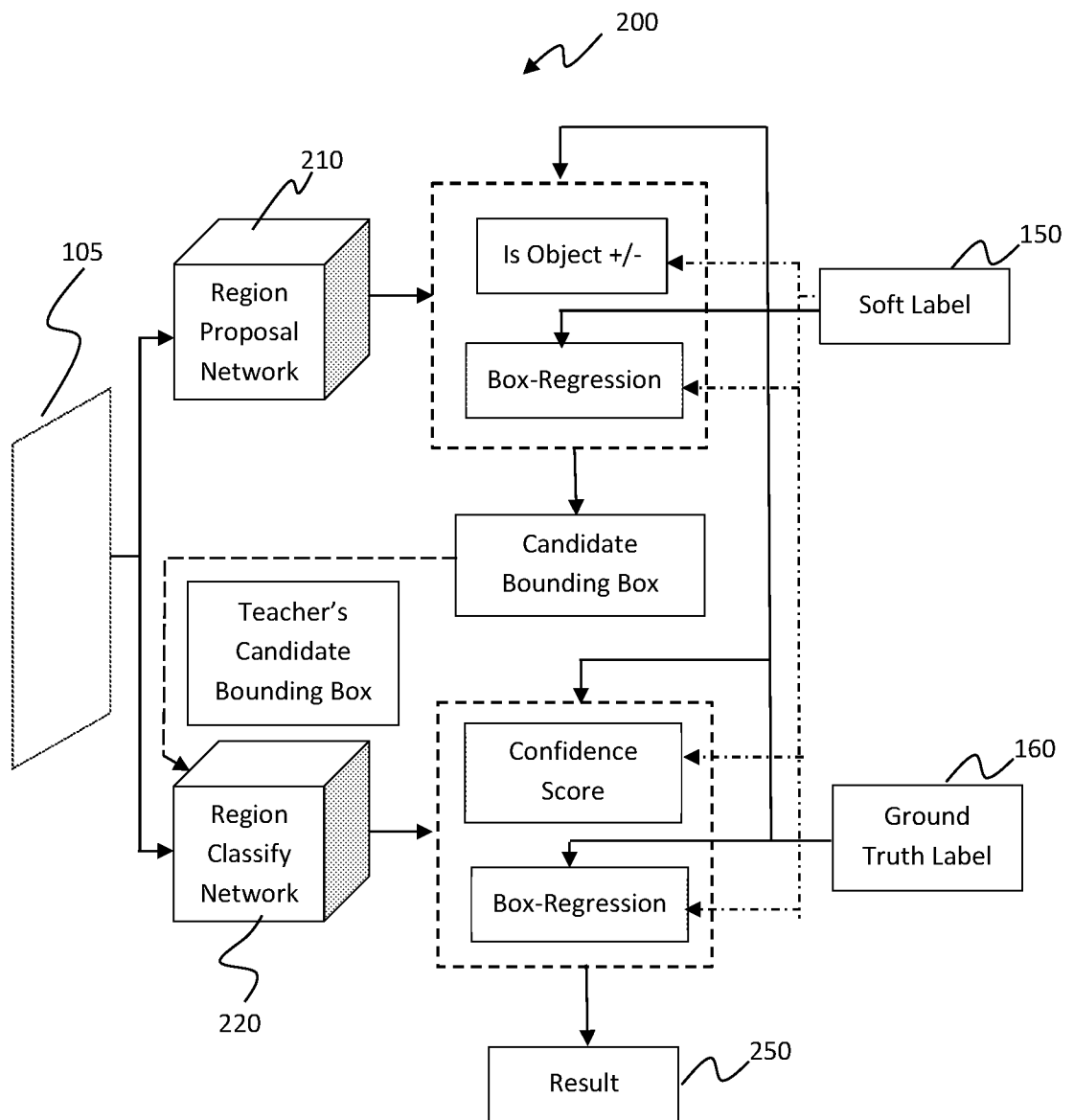


FIG. 2

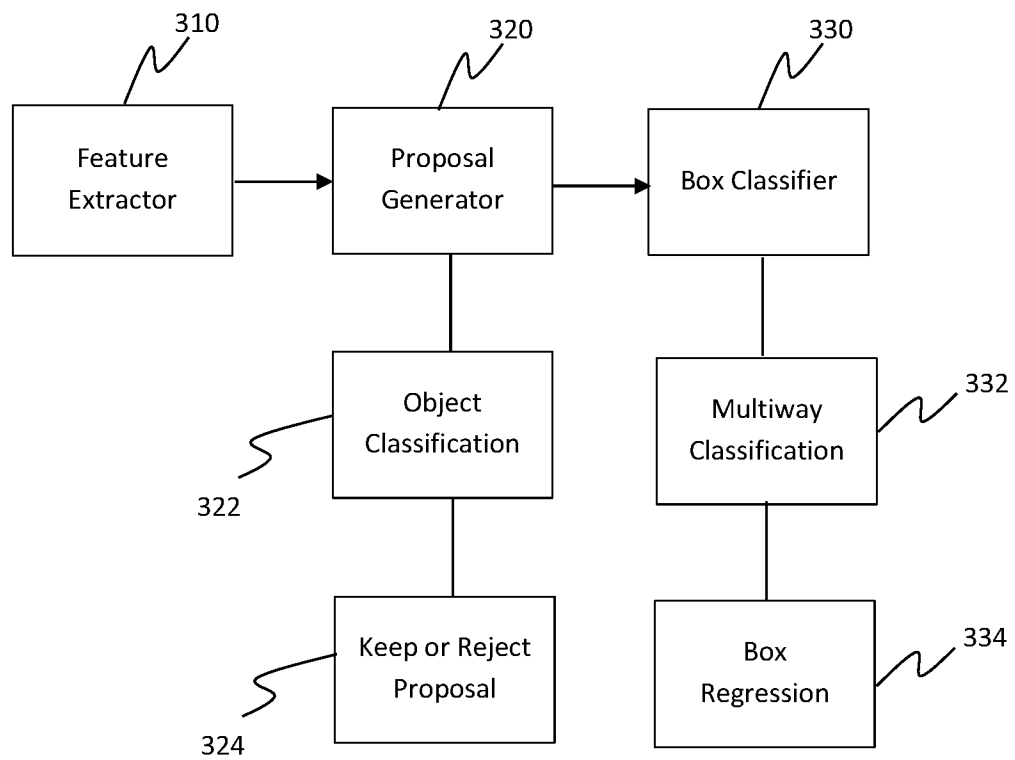
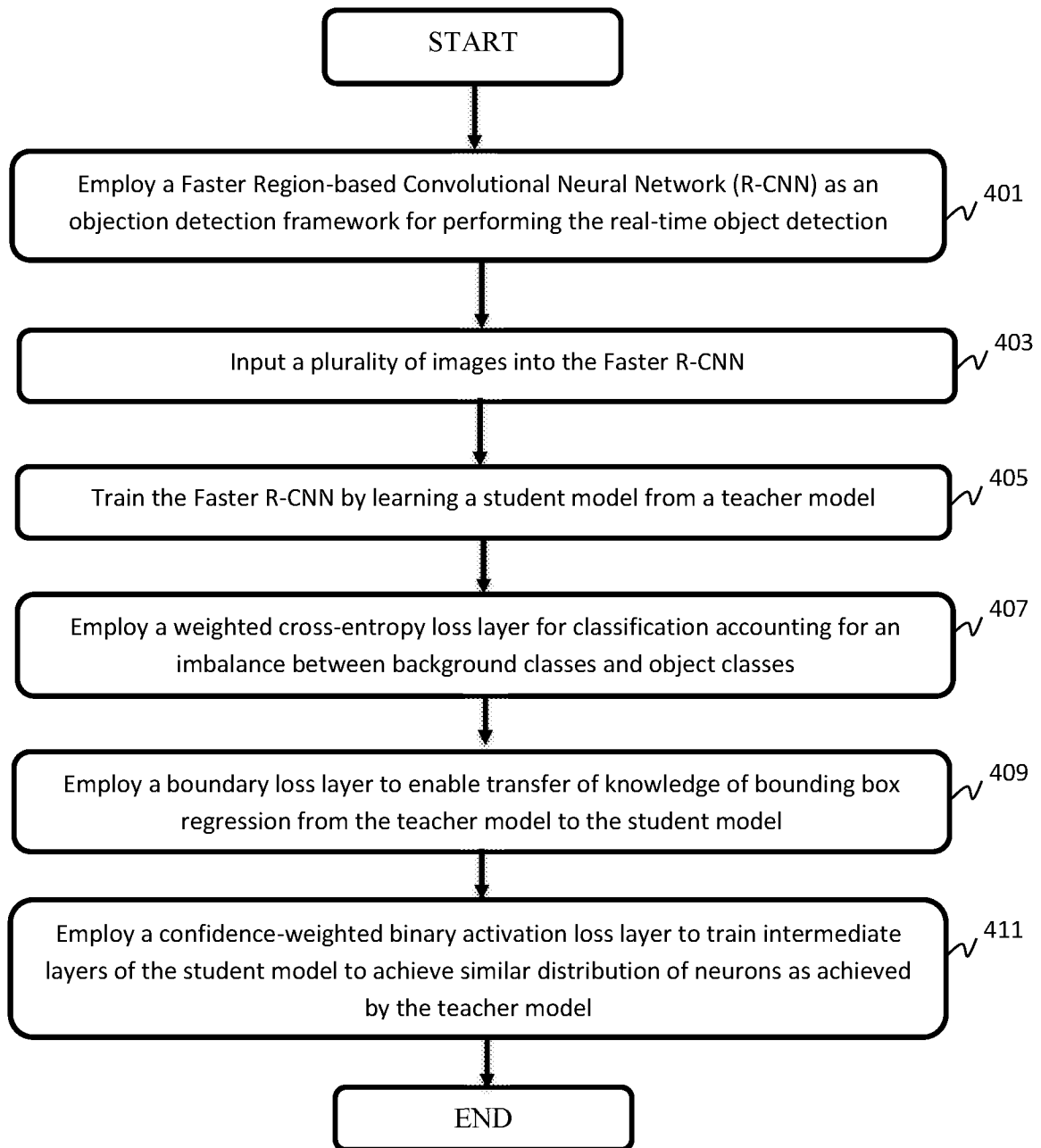


FIG. 3

**FIG. 4**

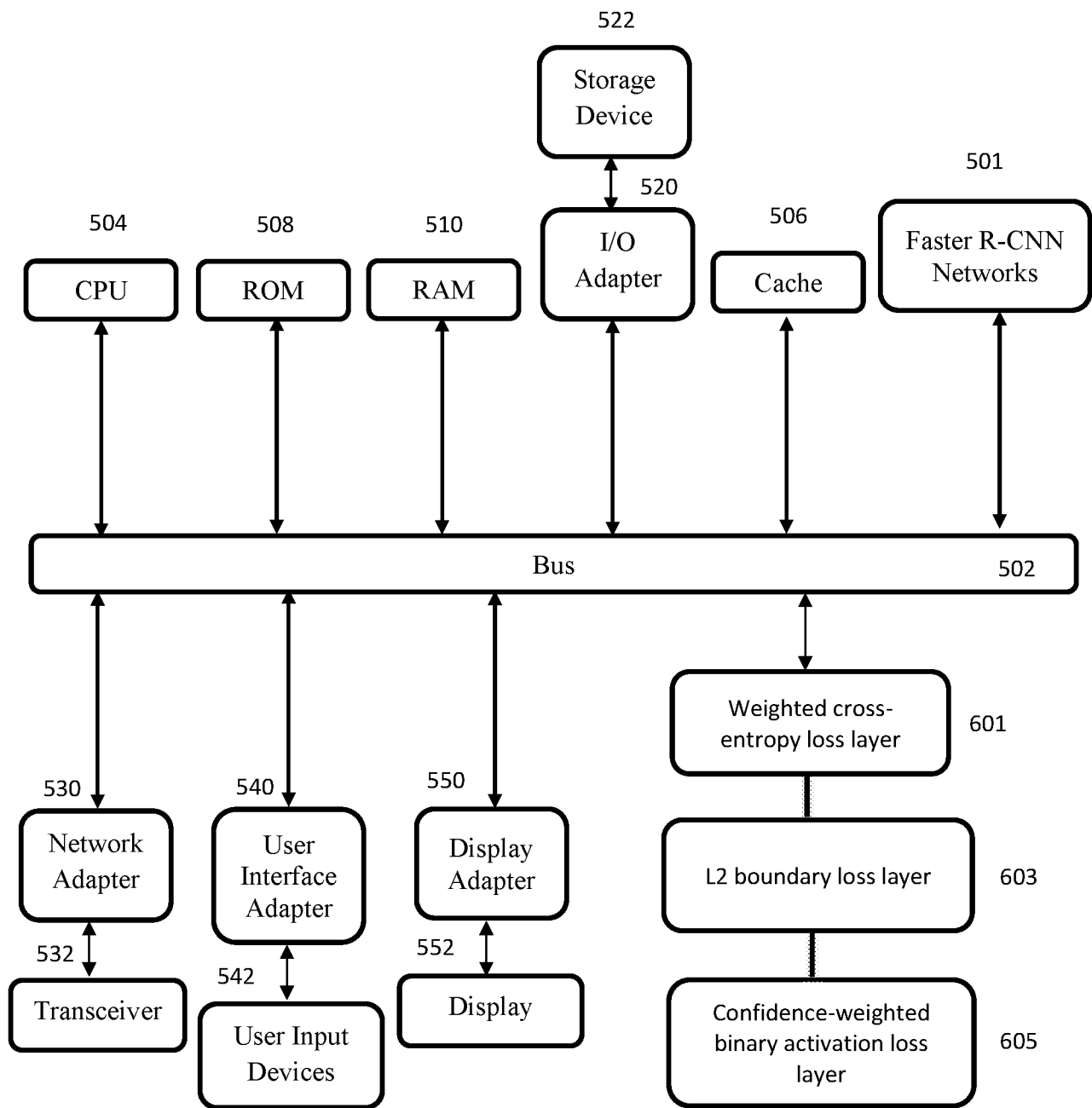


FIG. 5

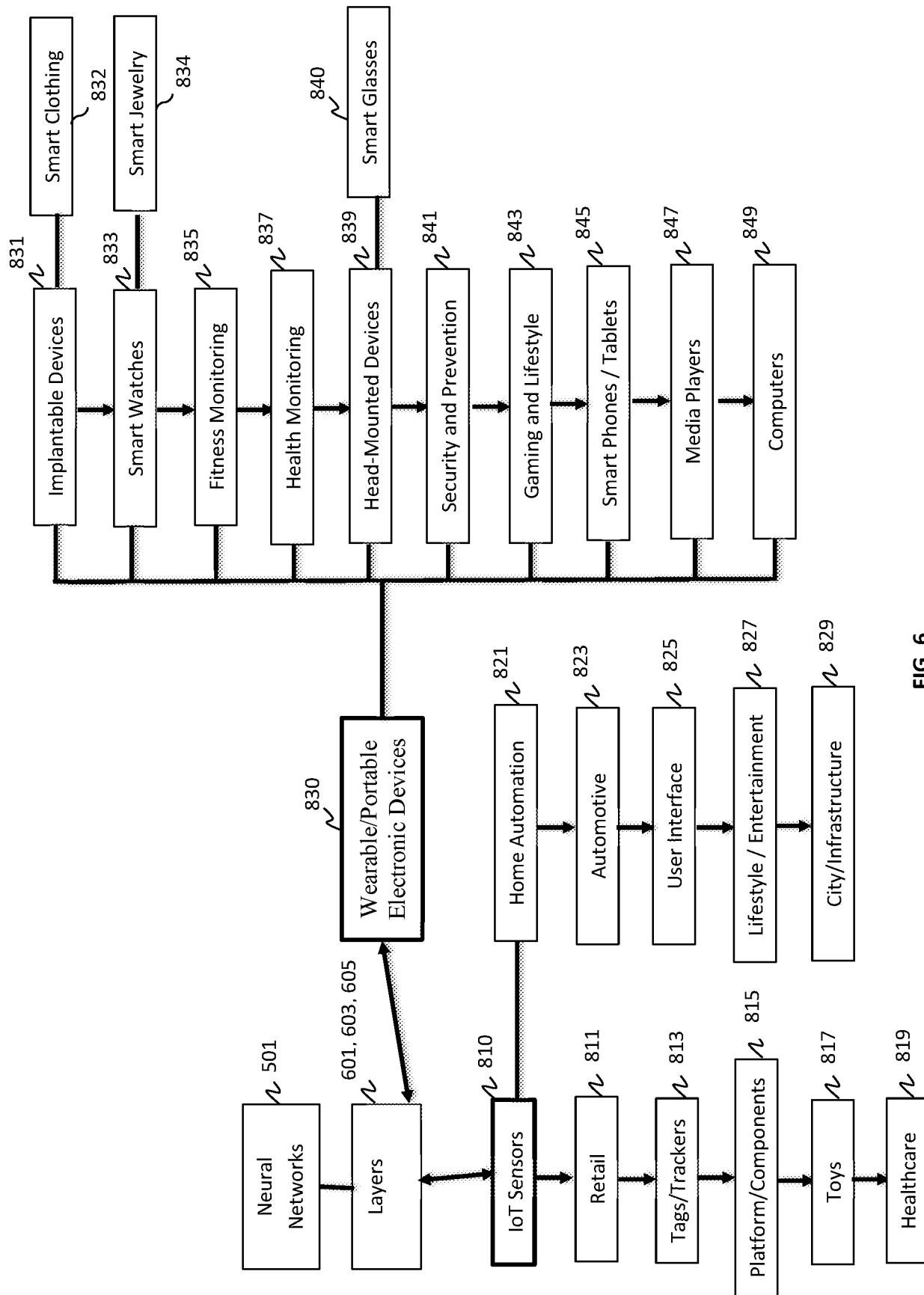


FIG. 6

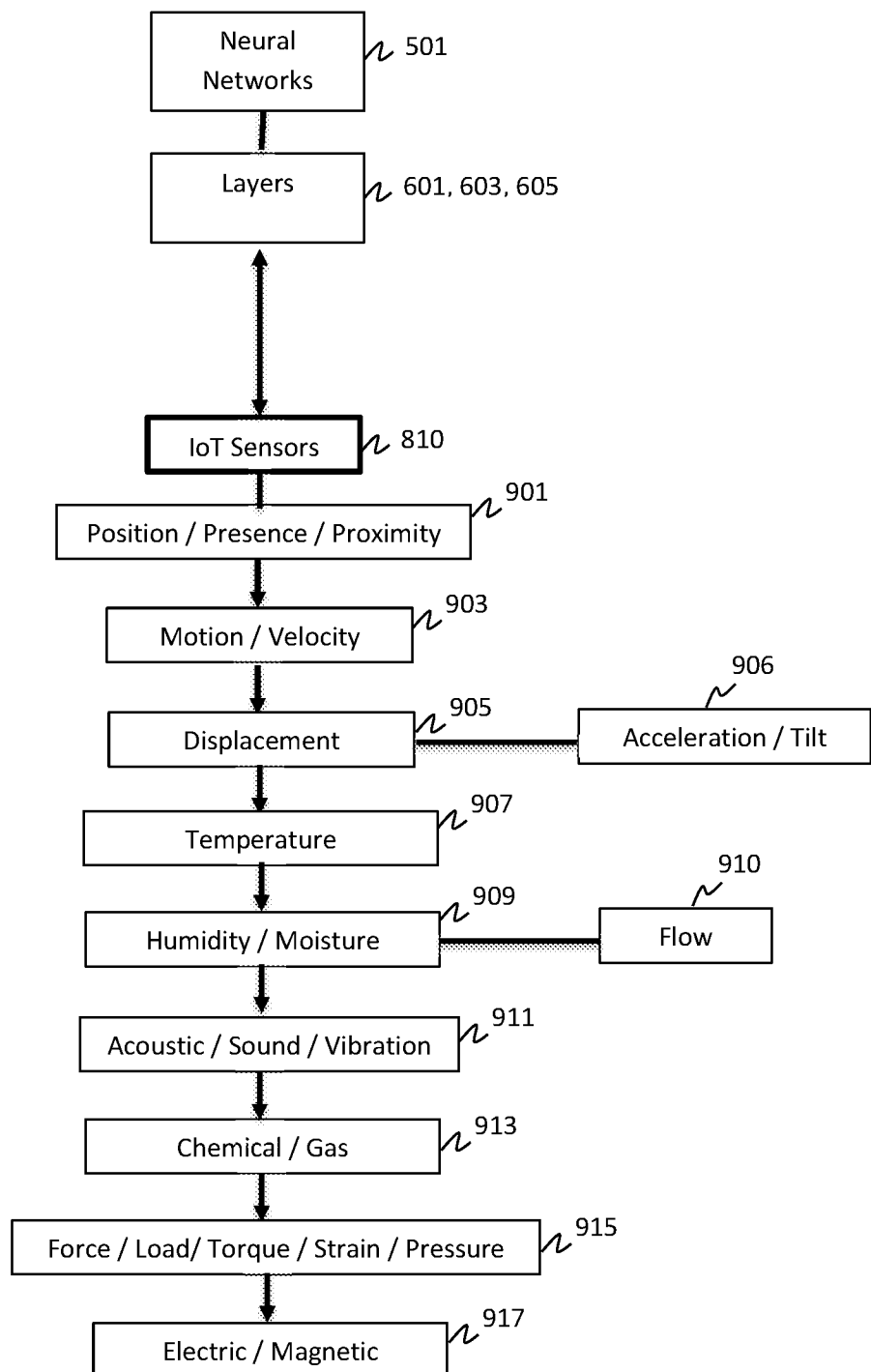


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2018/020863**A. CLASSIFICATION OF SUBJECT MATTER****G06N 3/08(2006.01)i, G06N 3/04(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N 3/08; G06E 1/00; G06K 9/66; G06K 9/62; G06K 9/00; G06N 3/04

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & keywords: Faster Region-based convolutional Neural Network (R-CNN), teacher/student model, hint-based learning, weighted cross-entropy loss layer, boundary loss layer, confidence-weighted binary activation loss layer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	SHAOQUING REN et al., `Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks`, arxiv.org, arXiv:1506.01497v3 [cs.CV], 06 January 2016, pages 1-14 Retrieved from <https://arxiv.org/abs/1506.01497> See pages 3-6.	1-20
Y	US 2016-0321522 A1 (CANON KABUSHIKI KAISHA) 03 November 2016 See paragraph [0095]; and claims 1, 4, 7, 11.	1-20
Y	ADRIANA ROMERO et al., `FITNETS: HINTS FOR THIN DEEP NETS`, arxiv.org, arXiv:1412.6550v4 [cs.LG], 27 March 2015, pages 1-13 Retrieved from <https://arxiv.org/abs/1412.6550> See pages 3-5, 8-9.	1-20
A	US 9262698 B1 (VICARIOUS FPC, INC.) 16 February 2016 See column 7, lines 46-57; claim 1; and figure 6.	1-20
A	US 2013-0259307 A1 (CANON KABUSHIKI KAISHA) 03 October 2013 See paragraphs [0032]-[0039]; claim 1; and figure 2.	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

15 June 2018 (15.06.2018)

Date of mailing of the international search report

15 June 2018 (15.06.2018)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2018/020863

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2016-0321522 A1	03/11/2016	None	
US 9262698 B1	16/02/2016	US 2016-0260009 A1 US 2016-0260010 A1 US 2016-0267375 A1 US 2017-0193374 A1 US 9373085 B1 US 9607262 B2 US 9607263 B2	08/09/2016 08/09/2016 15/09/2016 06/07/2017 21/06/2016 28/03/2017 28/03/2017
US 2013-0259307 A1	03/10/2013	JP 06032921 B2 JP 2013-210968 A US 9292745 B2	30/11/2016 10/10/2013 22/03/2016