(12)Indian Patent Application

---

(21) Application Number: 202141029134

(22) Filing Date: 29/06/2021 (43) Publication Date: 19/05/2023

(71) Applicant(s): L&T TECHNOLOGY SERVICES LIMITED

(72) Inventor(s): Kumar, Bharath Muniyappanavara
Ramachandrapurapu, Satya Karthik

(51) International Classifications: G06N 3/04 G06N 3/08 G06K 9/62 G06K 9/00 G06K 9/46

(54) Title: METHOD AND SYSTEM FOR TRAINING A CONVOLUTION NEURAL NETWORK (CNN) AND INFERENCING USING TRAINED CNN

(57) Abstract: A method and a system for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image is disclosed. The method comprises selecting one or more computer vision tasks to be performed on an image 'and performing a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. Said method further discloses dropping one or more features from the feature map and performing the one or more computer vision task on the image based on remaining features. Said method then compares result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses and updates weights based on a weighted sum loss calculated from the individual losses, which is used as feedback to the plurality of convolutional layers for training the CNN.

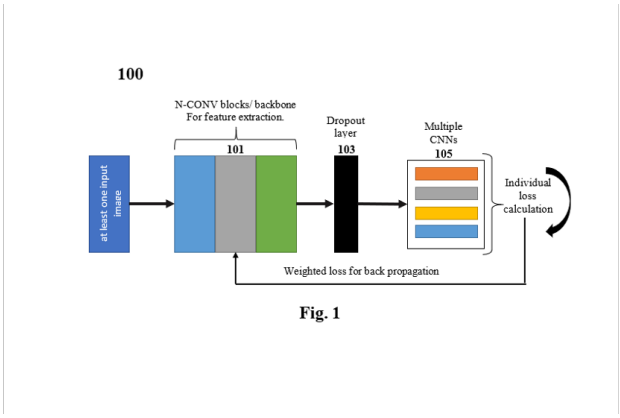

Fig. 1

FORM 2

THE PATENTS ACT 1970
(39 OF 1970)
&
The Patent Rules, 2003

Complete Specification

(See Section 10 and Rule 13)

1. TITLE OF THE INVENTION

METHOD AND SYSTEM FOR TRAINING A CONVOLUTION NEURAL NETWORK (CNN) AND INFERENCING USING TRAINED CNN

2. APPLICANT(S)

(a) NAME : **L&T TECHNOLOGY SERVICES LIMITED**

(b) NATIONALITY : **INDIAN**

(c) ADDRESS : **DLF IT SEZ Park, 2nd Floor – Block 3**

1/124, Mount Poonamallee Road,

Ramapuram, Chennai – 600 089,

INDIA.

3. PREAMBLE TO THE DESCRIPTION

COMPLETE

The following specification describes the invention and the manner in which it is to be performed

DESCRIPTION

Technical Field

[0001] The present disclosure generally relates to the field of computer vision. More specifically, it relates to training a convolution neural network (CNN) for performing one or more computer vision tasks on an image.

BACKGROUND

[0002] Computer Vision has gained importance in various application areas because it has given computers the ability "see" and understand the content of digital images such as photographs and videos. Computer vision is generally used for object classification, object identification, and object tracking.

[0003] Existing technique of training convolution neural network (CNN) for performing computer vision tasks involves different feature extraction network/layer for different tasks like object detection, segmentation, classification, etc., that are performed on the same image. Each of the extracted features are processed using individual network predictors and the individual loss for each task performed on the image is calculated independently.

[0004] For example, there can be two computer vision task that are of similar type. A first network is detecting vehicles on the road and a second network is detecting pedestrians. However, both these networks are performing similar task of predicting bounding boxes around the objects. In existing technique both these networks are trained independently.

[0005] The different feature extraction layers for each and every computer vision task performed on the same image may result in additional hardware requirement and also increases the cost of training neural networks, in terms of more GPU hours required to train the system. Further, the different feature extraction layers increases the time period for computer vision task in a real-time scenarios.

[0006] Therefore, there exists a need in the art to provide a method and a system which overcomes the above-mentioned problems by fast training and inferencing of the convolution neural networks (CNNs), thereby reducing the processing power required for computer vision related tasks.

OBJECTS OF THE PRESENT DISCLOSURE

[0007] Some of the objects of the present disclosure, which at least one embodiment herein satisfies are as listed herein below.

[0008] The primary object of the present disclosure is to train a convolution neural network (CNN) for performing one or more computer vision tasks on an image with increase efficiency.

[0009] Another object of the present disclosure is to reduced cost of training neural networks.

[0010] Yet another object of the present disclosure is to reduce the number of floating point operations (FLOPS) required by the hardware by combining the initial layers of CNN.

[0011] Yet another object of the present invention is to reduce the processing power required for computer vision related tasks.

SUMMARY

[0012] The present disclosure overcomes one or more shortcomings of the prior art and provides additional advantages discussed throughout the present disclosure. Additional features and advantages are realized through the techniques of the present disclosure. Other embodiments and aspects of the disclosure are described in detail herein and are considered a part of the claimed disclosure.

[0013] In one non-limiting embodiment of the present disclosure, a method for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image is disclosed. The method comprises selecting one or more computer vision tasks to be performed on an image and performing a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. Said method further discloses dropping one or more features from the feature map and performing the one or more computer vision task on the image based on remaining features of the feature map. Said method then discloses comparing result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses and updating weights based on a weighted sum loss calculated from the individual losses. Further, said method discloses applying said weights to the plurality of convolutional layers, as feedback, for training the CNN, the trained CNN being used for detecting the one or more objects in the image.

[0014] In another non-limiting embodiment of the present disclosure, the method further comprises selecting N-different sets of features from the feature map, performing the one or more computer vision tasks on the image based on each of the N-different sets of features, and comparing result of the each of the one or more computer vision tasks with respective ground truth result to calculate individual losses for each of the N-different sets of features. Said method further discloses updating weights based on a weighted sum loss calculated from the

individual losses for each of the N-different sets of features and applying the updated weights to the convolutional layers for training the CNN.

[0015] In yet another non-limiting embodiment of the present disclosure, the selection of the N-different sets of features from the feature map comprises dropping one or more features from the feature map for N-different sets of features, wherein the dropped one or more features are different for N-different sets.

[0016] In yet another non-limiting embodiment of the present disclosure, the individual calculated losses indicate differences between the calculated result of each of the one or more computer vision tasks and the respective ground truth result.

[0017] In yet another non-limiting embodiment of the present disclosure, the method further comprises capturing a real-time image for processing, performing a plurality of convolution operations on the captured image using the plurality of convolution layers of the trained CNN, to generate a feature map for the captured image, and performing one or more computer vision tasks on the captured image based on the feature map.

[0018] In yet another non-limiting embodiment of the present disclosure, a system for training a convolution neural network (CNN) to perform one or more computer vision tasks on an image is disclosed. The system comprises a memory, one or more processors coupled to the memory and configured to allow a selection unit to select one or more computer vision tasks to be performed on an image and a convolution unit to perform a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. The one or more processors are further configured to allow a dropout unit to drop one or more features from the feature map. The one or more processors are further configured to allow a task performance unit to perform the one or more computer vision task on the image based on remaining features of the feature map. The one or more processors are further configured to allow a comparison unit compare result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses, and a weighing unit to update weights based on a weighted sum loss calculated from the individual losses and apply said weights to the plurality of convolutional layers, as feedback, for training the CNN, wherein the trained CNN is used to perform the one or more computer vision tasks on the image.

[0019] In yet another non-limiting embodiment of the present disclosure, the one or more processors are further configured to select N-different sets of features from the feature map, perform, by the task performance unit, the one or more computer vision tasks on the image

based on each of the N-different sets of features, compare, by the comparison unit, result of the each of the one or more computer vision tasks with respective ground truth result to calculate individual losses for each of the N-different sets of features. Said one or more processors further configured to update, by the weighing unit, weights based on a weighted sum loss calculated from the individual losses for each of the N-different sets of features and apply, by the weighing unit, the updated weights to the convolutional layers to train the CNN.

[0020] In yet another non-limiting embodiment of the present disclosure, to select the N-different sets of features from the feature map, the one or more processors are configured to drop, by the dropout unit, one or more features from the feature map for N-different sets of features, wherein the dropped one or more features are different for N-different sets.

[0021] In yet another non-limiting embodiment of the present disclosure, an image capturing unit configured to capture a real-time image for processing. Said one or more processors are configured to perform, by the convolution unit, a plurality of convolution operations on the captured image using the plurality of convolution layers of the trained CNN, to generate a feature map for the captured image, and perform, by the task performance unit, one or more computer vision tasks on the captured image based on the feature map.

[0022] The foregoing summary is illustrative only and is not intended to be in any way limiting. In addition to the illustrative aspects, embodiments, and features described above, further aspects, embodiments, and features will become apparent by reference to the drawings and the following detailed description.

BRIEF DESCRIPTION OF THE ACCOMPANYING DRAWINGS

[0023] The features, nature, and advantages of the present disclosure will become more apparent from the detailed description set forth below when taken in conjunction with the drawings in which like reference characters identify correspondingly throughout. Some embodiments of system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

[0024] Fig. 1 shows an architecture for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure;

[0025] Fig. 2 shows an exemplary environment for training a convolution neural network (CNN) for traffic sign recognition (TSR) and lane detection, in accordance with an exemplary embodiment of the present disclosure;

[0026] Fig. 3 shows an exemplary environment for traffic sign recognition (TSR) and lane detection, in accordance with an embodiment of the present disclosure;

[0027] Fig. 4 shows a flow chart illustrating an exemplary method for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure;

[0028] Fig. 5 shows a flow chart illustrating an exemplary method for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure;

[0029] Fig. 6 shows a block diagram illustrating a system for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure.

[0030] It should be appreciated by those skilled in the art that any block diagram herein represents conceptual views of illustrative systems embodying the principles of the present subject matter. Similarly, it will be appreciated that any flow charts, flow diagrams and the like represent various processes which may be substantially represented in computer readable medium and executed by a computer or processor, whether or not such computer or processor is explicitly shown.

DETAILED DESCRIPTION

[0031] The terms “comprise”, “comprising”, “include(s)”, or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, system or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or system or method. In other words, one or more elements in a system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the system or apparatus.

[0032] In the following detailed description of the embodiments of the disclosure, reference is made to the accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure, and it is to be understood that other embodiments may be utilized and that changes may be made without departing from the scope of the present disclosure. The following description is, therefore, not to be taken in a limiting sense.

[0033] A method and a system for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image is disclosed. The method comprises selecting one or more computer vision tasks to be performed on an image `and performing a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. Said method further discloses dropping one or more features from the feature map and performing the one or more computer vision task on the image based on remaining features of the feature map. Said method then compares result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses and updates weights based on a weighted sum loss calculated from the individual losses, which is used as feedback to the plurality of convolutional layers for training the CNN.

[0034] **Fig. 1** shows an architecture 100 for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure.

[0035] In an embodiment of the present disclosure, at least one input image from one or more sources may be used for training the CNN for performing a number of computer vision tasks. The at least said image used for training the CNN may be provided to the CNN in batches. The number of computer vision tasks may involve object identification, segmentation, classification, etc. Further, said image may comprise multiple objects or entities.

[0036] In an exemplary embodiment, the objects or entities may include a pedestrian, a car, a traffic signal, an animal, a travelling lane, etc. It is to be noted said objects or entities are not limited to above examples and any other object or entity that may be detected using computer vision is well within the scope of the present disclosure. In one non-limiting embodiment of the present disclosure, the computer vision task may comprise face detection and posture estimation.

[0037] In an aspect of the present disclosure, a plurality of convolution operations are performed on the at least said image using a number of convolution layers 101 of the convolution neural network (CNN), to generate a feature map for the selected image. The feature map may comprise a plurality of features associated with a plurality of objects or entities presented in said image.

[0038] Further, the convolutions layers 101 of the CNN may be used to extract various features from the image. In the convolution layer 101, the number of convolution operations are performed between the input image and a filter of a particular size MxM. By sliding the filter

over the image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM) to generate an output.

[0039] The output of the convolution operations is termed as the feature map, which provides information about the image such as the corners and edges. The feature map is then fed to other layers of the CNN to learn several other features from said image. In one non-limiting embodiment of the present disclosure, ResNet architecture may be used for feature extraction. However, the feature extraction is not limited to above mentioned architecture and a person skilled in art may use any other architecture for feature extraction, which is known to a person skilled in the art. In an embodiment of the present disclosure, batch norm technique may be applied to the output of the convolution layers 101 for normalization. The CNN may comprise an activation function to learn and approximate any kind of continuous and complex relationship between variables of the network. The activation function may add non-linearity to the network.

[0040] It is to be appreciated that the above mentioned feature map of the image act as a common backbone for performing one or more computer vision tasks. By using a common backbone, the CNN learns the relationship between different features associated with different objects or entities, the number of floating point operations performed by the CNN is reduced and speed of training and inferencing is improved.

[0041] Said architecture 100 further includes a dropout layer 103 being connected to the convolution layers 101. In an embodiment of the present disclosure, the dropout layer 103 of the said architecture 100 is configured to randomly drop one or more features from the feature map by dropping few neurons of the neural network.

[0042] The dropout layer 103 ensures that the CNN does not memorizes a specific pattern. The dropout layer 103 further helps in training the CNN to perform one or more computer vision tasks with lesser number of features than actual number of features available, thereby improving the efficiency of the CNN. In one non-limiting embodiment of the present disclosure, the dropout layer 103 may be a part of the CNN.

[0043] Said architecture 100 further comprises multiple CNNs 105 that are configured to select the remaining features of the feature map and perform one or more computer vision tasks based on the selected remaining features to generate result for each of the one or more computer vision tasks.

[0044] Said architecture 100 may further include a comparator (not shown) that is configured to compare result of each of the one or more computer vision tasks with respective ground

truth result to calculate individual losses of the one or more computer vision tasks. In one non-limiting embodiment, the individual losses may be associated with segmentation, bounding boxes, etc. It is to be noted said losses are not limited to above examples and any other loss calculated by comparing the result and ground truth result is well within the scope of the present disclosure.

[0045] In another aspect of the present disclosure, said architecture 100 may further include a weighing unit (not shown) configured to update weights based on a weighted sum loss calculated from the individual losses and these updated weights may be applied as a feedback to the convolutional layers 101 to improve the result of the convolution operations. The updated weights may compensate for the calculated individual losses in the result of the one or more computer vision tasks. In a second iteration, the convolution operations are then performed based on the applied updated weights to generate a feature map.

[0046] The dropout layer 103 may be again used to randomly drop one or more features. The dropped features are different in each iteration. The one or more computer vision tasks are then performed based on the remaining of the features and results are compared with ground truth result to calculate updated weights for training the CNN as discussed above. The CNN may be trained with N-number of iteration to minimize the loss to the smallest possible value and to efficiently perform one or more computer vision tasks on the image.

[0047] In an embodiment of the present disclosure, during the inference process the dropout layer and the feedback loop are removed. The trained CNN may be used for performing one or more computer vision tasks on the image using a common backbone or shared initial layers, thereby reducing the processing power requirement for the detection.

[0048] Fig. 2 shows an exemplary environment for training a convolution neural network (CNN) for traffic sign recognition (TSR) and lane detection, in accordance with an embodiment of the present disclosure.

[0049] In an embodiment of the present disclosure, at least one image captured by an image sensor of the automobile is used for training a convolution neural network (CNN). The captured at least one image may be used for training the CNN for one or more computer vision tasks such as driving lane detection and traffic signal recognition (TSR).

[0050] Initially, a plurality of convolution operations are applied on said at least one image using a plurality of convolutional layers 101 to generate a single or common feature map for lane detection and TSR. By using single shared feature map, the network may also learn the relationship between different features.

[0051] The dropout layer 103 is used to randomly drop one or more features of the feature map and the TSR and lane predictor networks 105 may predict the driving lane and traffic signal based on remaining of the features left after dropping. The dropout layer 103 facilitates training of the CNN with lesser number of features than actual number of features available, thereby improving the efficiency of the CNN in performing one or more computer vision tasks.

[0052] The predicted results of the TSR and lane predictor networks 105 are compared with respective ground truth result to calculate the individual losses of traffic signal and driving lane. Then a weighted sum loss is calculated from the individual losses and the weighted sum loss is used to update the weights. The updated weights may be applied as a feedback to the convolutional layers 101 to improve the convolution operation and feature extraction.

[0053] The convolution operations are again performed on the image based on the applied updated weights to generate a feature map. The dropout layer 103 may drop one or more features of the feature map as discussed above. The one or more features dropped in the second iteration are different from the features dropped at the initial stage and are selected randomly in every iteration.

[0054] In one non-limiting embodiment of the present disclosure, the dropout layer 103 may drop 20/30/40 percent of the extracted features. It is to be noted said dropping of features is not limited to above percentage and any other drop percentage selection is well within the scope of the present disclosure.

[0055] The TSR and lane predictor networks 105 may comprise the multiple CNNs for predicting the traffic signal and driving lane. The predicted traffic signal and driving lane is compared with the original traffic signal and driving lane to calculate individual losses of traffic signal and driving lane. The individual losses may be used to calculate a weighted sum loss. The weighted sum loss is used to update weights and apply the updated weights to the convolutional layers 101 for improving the feature extraction, using the procedure discussed above.

[0056] The application of updated weights and individual loss calculation is carried out iteratively for N- number of times to train the CNN and improve the prediction result. The CNN is thus trained with N-number of iteration to minimize the loss to the smallest possible value and to efficiently predict the object traffic signal and driving lane present in the image in said example.

[0057] **Fig. 3** shows an exemplary environment for traffic signal recognition (TSR) and lane detection, in accordance with an embodiment of the present disclosure.

[0058] In an embodiment of the present disclosure, the trained CNN discussed in above embodiments is used for TSR and lane detection in an image. The image is captured by an image sensor of the automobile in real-time while travelling on the road. A plurality of convolution operations are then performed on the captured image using the convolution layers to generate a feature map for the received image.

[0059] The individual CNN i.e. the TSR network and lane network shall predict/detect the traffic signal and driving lane in the captured image based on the extracted features. The output of TSR network may be bonding boxes and output of the lane network may be segmentation. The initial shared layer for feature extraction reduces the processing power requirement and reduces the latency in the one or more computer vision tasks. The CNN trained with a lesser number of features than available can now predict the traffic signal and driving lane more efficiently with all the features available.

[0060] **Fig. 4** shows a flow chart illustrating an exemplary method 400 for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure.

[0061] At block 402, the method 400 discloses selecting one or more computer vision tasks to be performed on an image. The image may be used to train a convolution neural network (CNN) for performing one or more computer vision tasks, as discussed in above embodiments. In one non-limiting embodiment, the CNN may be trained with a plurality of image for performing one or more computer vision tasks.

[0062] In one non-limiting embodiment of the present disclosure, the one or more computer vision tasks may comprise object detection, face detection, and posture estimation. In another non-limiting embodiment of the present disclosure, the object identification may comprise traffic sign recognition and lane detection. It is to be noted that the one or more computer vision tasks is not limited to above examples and any other computer task is well within the scope of the present disclosure.

[0063] The at least one image used for training may be stored locally or remotely. In one non-limiting embodiment of the present disclosure, the at least one image may be stored on a server and retrieved from the server for training. The at least one image may be present in any format known to a person skilled in the art.

[0064] At block 404, the method 400 discloses performing a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. The feature map may comprise a plurality of features associated with a plurality of objects/entities present within the image.

[0065] It is to be appreciated that the above mentioned shared feature map for plurality of entities/objects act as a common backbone for performing one or more computer vision tasks. By using a common backbone, the CNN learns the relationship between different features associated with different objects/entities. Further, the number of floating point operations performed by the CNN is reduced and speed of training and inferencing is also improved.

[0066] At block 406, the method 400 discloses dropping one or more features from the feature map and performing the one or more computer vision task on the image based on remaining features of the feature map. The features to be dropped may be selected randomly for every iteration such that the dropped feature of every iteration are different from each other.

[0067] The remaining of the features after dropping are used for performing one or more selected computer vision tasks. The CNN is thus trained in such a manner that the CNN performs one or more selected computer vision tasks on the image with lesser number of features than the actual number of features, thereby improving the efficiency of the CNN.

[0068] At block 408, the method 400 discloses comparing result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses. The individual calculated losses indicate differences between the calculated result of each of the one or more computer vision tasks and the respective ground truth result. In one non-limiting embodiment, the loss may be associated with the segmentation or bounding boxes, or any other loss calculated by comparing the result of computer vision task with and ground truth result.

[0069] At block 412, the method 400 discloses updating weights based on a weighted sum loss calculated from the individual losses. At block 412, the method 400 discloses applying the updated weights to the plurality of convolutional layers, as feedback, for training the CNN. The trained CNN may be used to perform the one or more computer vision tasks on the image. In one non-limiting embodiment, the updated weights may be applied to kernel or filter of the convolution layers for improving the result of the convolution operations.

[0070] In an embodiment of the present disclosure, the method 400 further discloses selecting N-different sets of features from the feature map, performing the one or more computer vision tasks on the image based on each of the N-different sets of features; comparing result of the

each of the one or more computer vision tasks with respective ground truth result to calculate individual losses for each of the N-different sets of features, updating weights based on a weighted sum loss calculated from the individual losses for each of the N-different sets of features, and applying the updated weights to the convolutional layers for training the CNN.

[0071] In an essential embodiment, the selection of the N-different sets of features from the feature map comprises dropping one or more features from the feature map for N-different sets of features and the dropped one or more features are different for N-different sets. It is to be thus noted that the computer vision tasks are performed based on the remaining features left after dropping certain features, which may be different each time, for every iteration. In this manner, the CNN may be trained with N-number of iteration to minimize the loss to the smallest possible value and to efficiently perform one or more computer vision tasks on the image.

[0072] In another embodiment of the present disclosure, the steps of method 400 may be performed in an order different from the order described above.

[0073] **Fig. 5** shows a flow chart illustrating an exemplary method 500 for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure.

[0074] The method 500 discloses an inferencing technique for performing one or more computer vision tasks on the image. At block 502, the method 500 discloses capturing a real-time image for processing. The image may be captured using one or more image sensors.

[0075] At block 504, the method 500 discloses performing a plurality of convolution operations on the captured image using the plurality of convolution layers of the trained CNN, to generate a feature map. The generated feature map may comprise features associated with one or more objects/entities present in the image.

[0076] At block 506, the method 500 discloses performing one or more computer vision tasks on the captured image based on the feature map using the trained CNN. As the CNN is trained with a lesser number of features than available, the trained CNN can now perform one or more computer vision tasks more efficiently with all the features available.

[0077] In another embodiment of the present disclosure, the steps of method 500 may be performed in an order different from the order described above.

[0078] Fig. 6 shows a block diagram illustrating a system 600 for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, in accordance with an embodiment of the present disclosure.

[0079] In an embodiment of the present disclosure, the system 600 may comprise an I/O interface 602, one or more processors 604, memory 606, and one or more convolution neural networks 608 communicatively coupled with each other. The system may further comprise units 610. The units 612 may include a selection unit 612, a convolution unit 614, a dropout unit 616, a task performance unit 618, a comparison unit 620, a weighing unit 622, and an image capturing unit 624 in communication with each other.

[0080] In an embodiment of the present disclosure, the selection unit 612, using the one or more processors 604, may be configured to select one or more computer vision tasks to be performed on an image. The image is used to train a convolution neural network (CNN) for one or more computer vision tasks discussed above.

[0081] The image used for training may be taken from one or more sources 650 that may be present locally or remotely to the system 600. In one non-limiting embodiment of the present disclosure, the at least one image may be stored on a server and retrieved from the server for training. The at least one image may be present in any format known to a person skilled in the art.

[0082] In one non-limiting embodiment of the present disclosure, the one or more computer vision tasks may comprise object detection, face detection, and posture estimation. In another non-limiting embodiment of the present disclosure, the object identification may comprise traffic sign recognition and lane detection. It is to be noted that the one or more computer vision tasks is not limited to above examples and any other computer task is well within the scope of the present disclosure.

[0083] The convolution unit 614, using the one or more processors 604, may be configured to perform a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. The feature map may comprise a plurality of features associated with a plurality of objects/entities.

[0084] It is to be appreciated that the above mentioned shared feature map for plurality of entities/objects act as a common backbone for performing one or more computer vision tasks. By using a common backbone, the CNN 608 learns the relationship between different features associated with different objects/entities. Further, the number of floating point operations performed by the CNN 608 is reduced and speed of training and inferencing is also improved.

[0085] The dropout unit 616, using the one or more processors 604, may be configured to drop one or more features from the feature map and the task performance unit 618 may be configured to perform the one or more computer vision task on the image based on remaining

features of the feature map. It is to be noted that the features to be dropped may be selected randomly for every iteration such that the dropped feature of every iteration are different from each other.

[0086] The remaining of the features after dropping are used to perform one or more computer vision tasks on the image. The CNN 608 is thus trained in such a manner that the CNN 608 performs one or more selected computer vision tasks on the image with lesser number of features than the actual number of features, thereby improving the efficiency of the CNN 608.

[0087] The comparison unit 622, using the one or more processors 604, may be configured to compare result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses. The individual calculated losses indicate differences between the calculated result of each of the one or more computer vision tasks and the respective ground truth result. In one non-limiting embodiment, the loss may be associated with the segmentation or bounding boxes, or any other loss calculated by comparing the result of computer vision task with and ground truth result.

[0088] The weighing unit 622 may be then configured to update weights based on a weighted sum loss calculated from the individual losses and apply said updated weights to the plurality of convolutional layers, as feedback, for training the CNN 608. The trained CNN 608 is used to perform the one or more computer vision operations on the image. In one non-limiting embodiment, the updated weights may be applied to kernel or filter of the convolution layers for improving the result of the convolution operations.

[0089] In an embodiment of the present disclosure, the one or more processors 604 may be further configured to select N-different sets of features from the feature map, perform, by the task performance unit 618, the one or more computer vision tasks on the image based on each of the N-different sets of features, compare, by the comparison unit 620, result of the each of the one or more computer vision tasks with respective ground truth result to calculate individual losses for each of the N-different sets of features, update, by the weighing unit 622, weights based on a weighted sum loss calculated from the individual losses for each of the N-different sets of features, and apply said updated weights, by the weighing unit 622, to the convolutional layers for training the CNN.

[0090] In an embodiment of the present disclosure, to select the N-different sets of features from the feature map, the one or more processors may be configured to drop, by the dropout unit 616, one or more features from the feature map for N-different sets of features and the dropped one or more features are different for N-different sets.

[0091] In an embodiment of the present disclosure, the system 600 may be used for performing one or more computer vision tasks on an image using the trained CNN 608. The system 600 may comprise the image capturing unit 624 configured to capture a real-time image for processing. The image capturing unit 624 may comprise one or more image sensors.

[0092] The one or more processors 604 may be configured to perform a plurality of convolution operations on the captured image using the plurality of convolution layers of the trained CNN 608 to generate a feature map. The generated feature map may comprise features associated with one or more objects/entities present in the image.

[0093] The CNN 608 trained discussed in above embodiments may be used to perform one or more computer vision tasks on the image based on the feature map. As the CNN is trained with a lesser number of features than available, the trained CNN can now perform one or more computer vision tasks on the image more efficiently with all features available.

[0094] In an embodiment, the units 612-622 may be dedicated hardware units capable of executing one or more instructions stored in the memory 606 for performing the operations of the system 600. In another embodiment, the units 612-622 may be software modules stored in the memory 506 which may be executed by the one or more processors 604 for performing the operations of the system 600.

[0095] The illustrated steps are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

[0096] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The

term “computer- readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., are non-transitory. Examples include random access memory (RAM), read-only memory (ROM), volatile memory, nonvolatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[0097] Suitable processors include, by way of example, a processor, a special purpose processor, a conventional processor, a digital signal processor (DSP), a plurality of microprocessors, one or more microprocessors in association with a DSP core, a controller, a microcontroller, Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs) circuits, any other type of integrated circuit (IC), and/or a state machine.

ADVANTAGES OF THE PRESENT DISCLOSURE

[0098] Exemplary embodiments discussed above may provide certain advantages. Though not required to practice aspects of the disclosure, these advantages may include those provided by the following features.

[0099] In an embodiment, the present disclosure trains a convolution neural network (CNN) for performing one or more computer vision tasks on the same image with increase efficiency.

[0100] In an embodiment, the present disclosure reduces reduced cost of training neural networks.

[0101] In an embodiment, the present disclosure reduces the number of floating point operations (FLOPS) required by the hardware by combining the initial layers of CNN.

[0102] In an embodiment, the present disclosure reduces the processing power required for computer vision related tasks.

WE CLAIM:

1. A method for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image, the method comprising:

selecting one or more computer vision tasks to be performed on an image;

performing a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image;

dropping one or more features from the feature map and performing the one or more computer vision task on the image based on remaining features of the feature map;

comparing result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses;

updating weights based on a weighted sum loss calculated from the individual losses; and

applying said weights to the plurality of convolutional layers, as feedback, for training the CNN, wherein the trained CNN is used for performing the one or more computer vision tasks on the image.

2. The method as claimed in claim 1, further comprising:

selecting N-different sets of features from the feature map;

performing the one or more computer vision tasks on the image based on each of the N-different sets of features;

comparing result of the each of the one or more computer vision tasks with respective ground truth result to calculate individual losses, for each of the N-different sets of features;

updating weights based on a weighted sum loss calculated from the individual losses for each of the N-different sets of features; and

applying the updated weights to the convolutional layers for training the CNN.

3. The method as claimed in claim 2, wherein selecting the N-different sets of features from the feature map comprises:

dropping one or more features from the feature map for N-different sets of features, wherein the dropped one or more features are different for N-different sets.

4. The method as claimed in claim 1, wherein the individual calculated losses indicate differences between the calculated result of each of the one or more computer vision tasks and the respective ground truth result.

5. The method as claimed in claim 1, further comprising:

capturing a real-time image for processing;

performing a plurality of convolution operations on the captured image using the plurality of convolution layers of the trained CNN, to generate a feature map for the captured image; and

performing one or more computer vision tasks on the captured image based on the feature map.

6. A system for training a convolution neural network (CNN) to perform one or more computer vision tasks on an image, the system comprising:

a memory;

one or more processors coupled to the memory and configured to allow:

a selection unit to select one or more computer vision tasks to be performed on an image;

a convolution unit to perform a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image;

a dropout unit to drop one or more features from the feature map;

a task performance to perform the one or more computer vision task on the image based on remaining features of the feature map;

a comparison unit to compare result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses; and

a weighing unit to update weights based on a weighted sum loss calculated from the individual losses and apply said weights to the plurality of convolutional layers, as feedback, for training the CNN, wherein the trained CNN is used to perform the one or more computer vision tasks on the image.

7. The system as claimed in claim 6, wherein the one or more processors are further configured to:

select N-different sets of features from the feature map;

perform, by the task performance unit, the one or more computer vision tasks on the image based on each of the N-different sets of features;

compare, by the comparison unit, result of the each of the one or more computer vision tasks with respective ground truth result to calculate individual losses for each of the N-different sets of features;

update, by the weighing unit, weights based on a weighted sum loss calculated from the individual losses, for each of the N-different sets of features; and

apply, by the weighing unit, the updated weights to the convolutional layers to train the CNN.

8. The system as claimed in claim 7, wherein to select the N-different sets of features from the feature map, the one or more processors are configured to:

drop, by the dropout unit, one or more features from the feature map for N-different sets of features, wherein the dropped one or more features are different for N-different sets.

9. The system as claimed in claim 6, wherein the individual calculated losses indicate differences between the calculated result of each of the one or more computer vision tasks and the respective ground truth result.

10. The system as claimed in claim 6, further comprising:

an image capturing unit configured to:

capture a real-time image for processing; and

wherein the one or more processors are configured to:

perform, by the convolution unit, a plurality of convolution operations on the captured image using the plurality of convolution layers of the trained CNN, to generate a feature map for the captured image; and

perform, by the task performance unit, one or more computer vision tasks on the captured image based on the feature map.

Dated this 24th day of June 2021

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

ABSTRACT

METHOD AND SYSTEM FOR TRAINING A CONVOLUTION NEURAL NETWORK (CNN) AND INFERENCING USING TRAINED CNN

A method and a system for training a convolution neural network (CNN) for performing one or more computer vision tasks on an image is disclosed. The method comprises selecting one or more computer vision tasks to be performed on an image `and performing a plurality of convolution operations on the image using a plurality of convolution layers to generate a feature map for the image. Said method further discloses dropping one or more features from the feature map and performing the one or more computer vision task on the image based on remaining features. Said method then compares result of each of the one or more computer vision tasks with respective ground truth result to calculate individual losses and updates weights based on a weighted sum loss calculated from the individual losses, which is used as feedback to the plurality of convolutional layers for training the CNN.

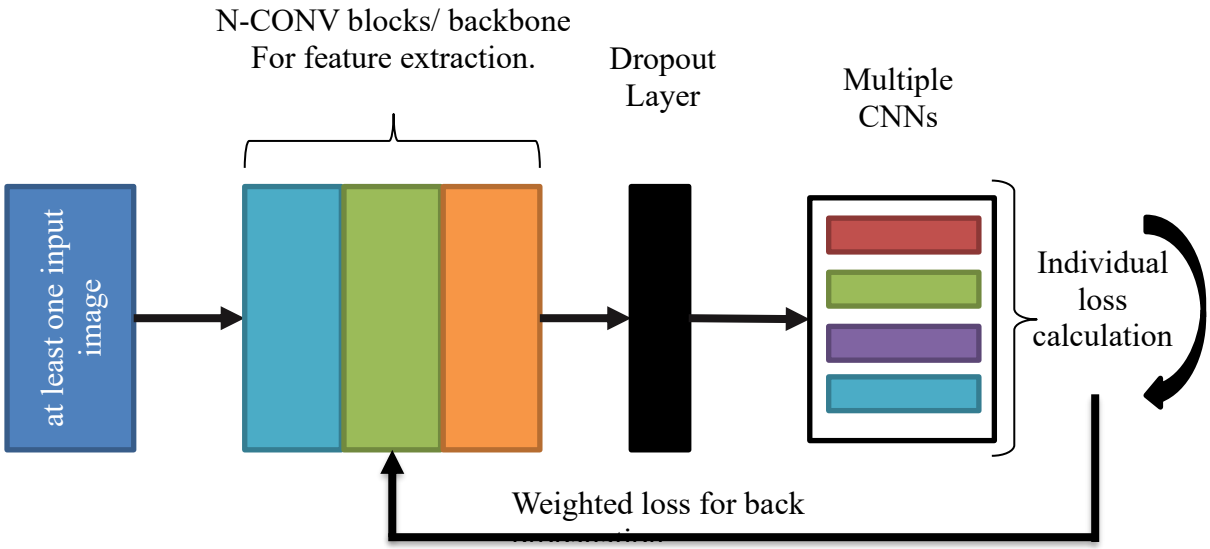

Fig. 1

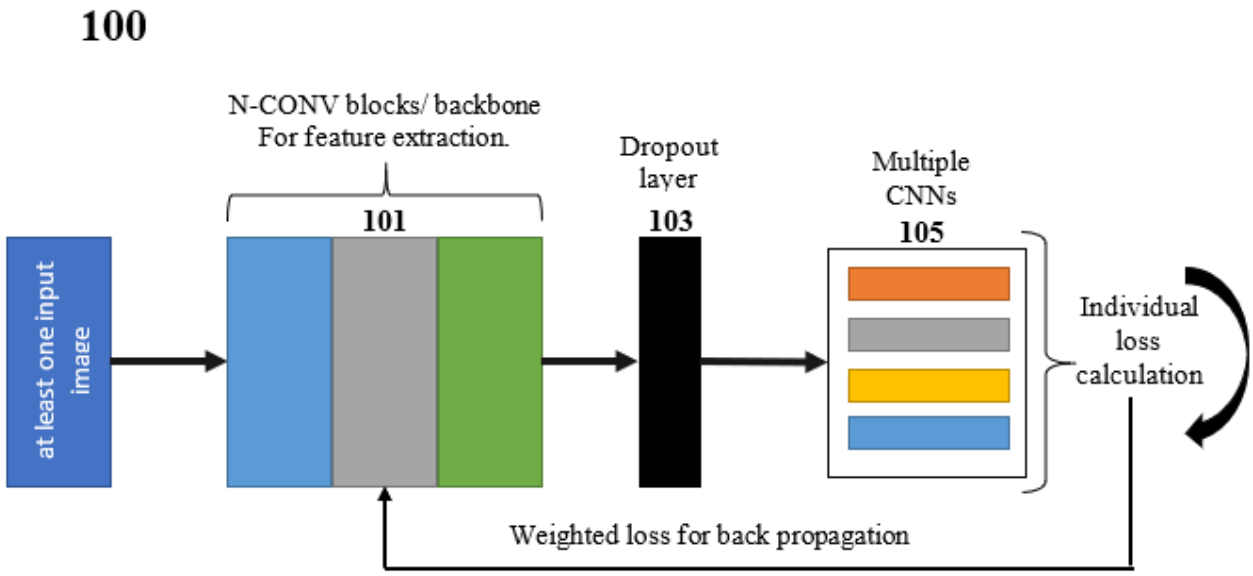

Fig. 1

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

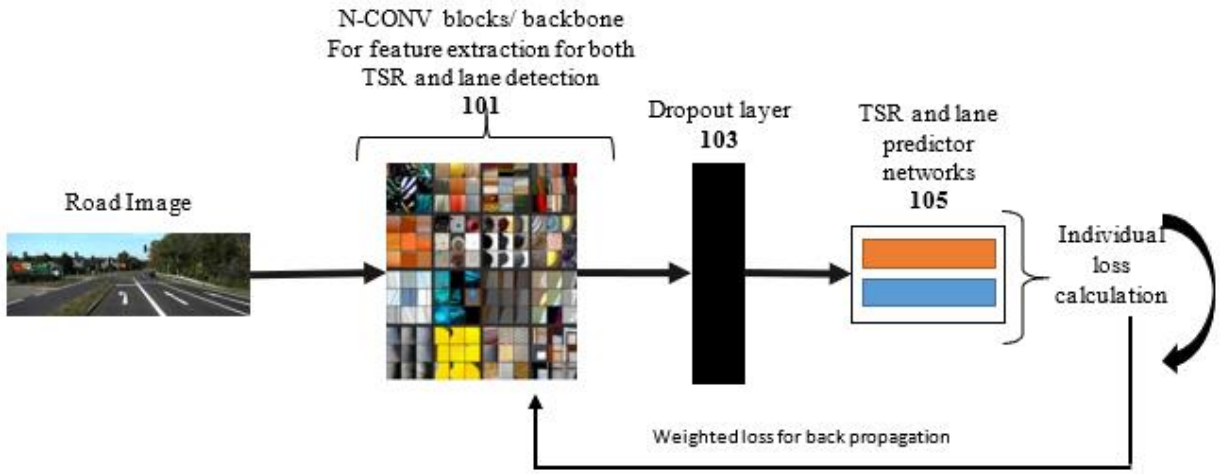

Fig. 2

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

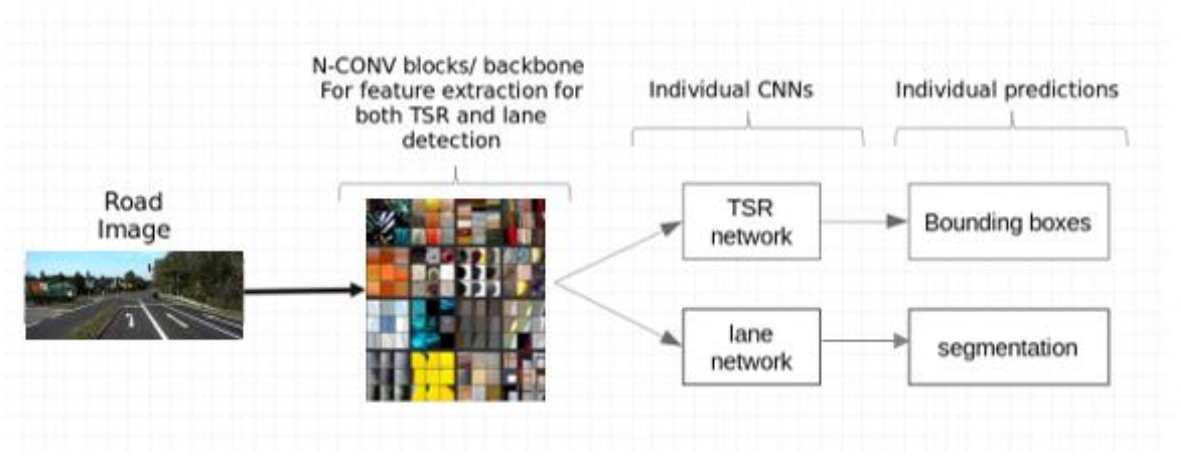

Fig. 3

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

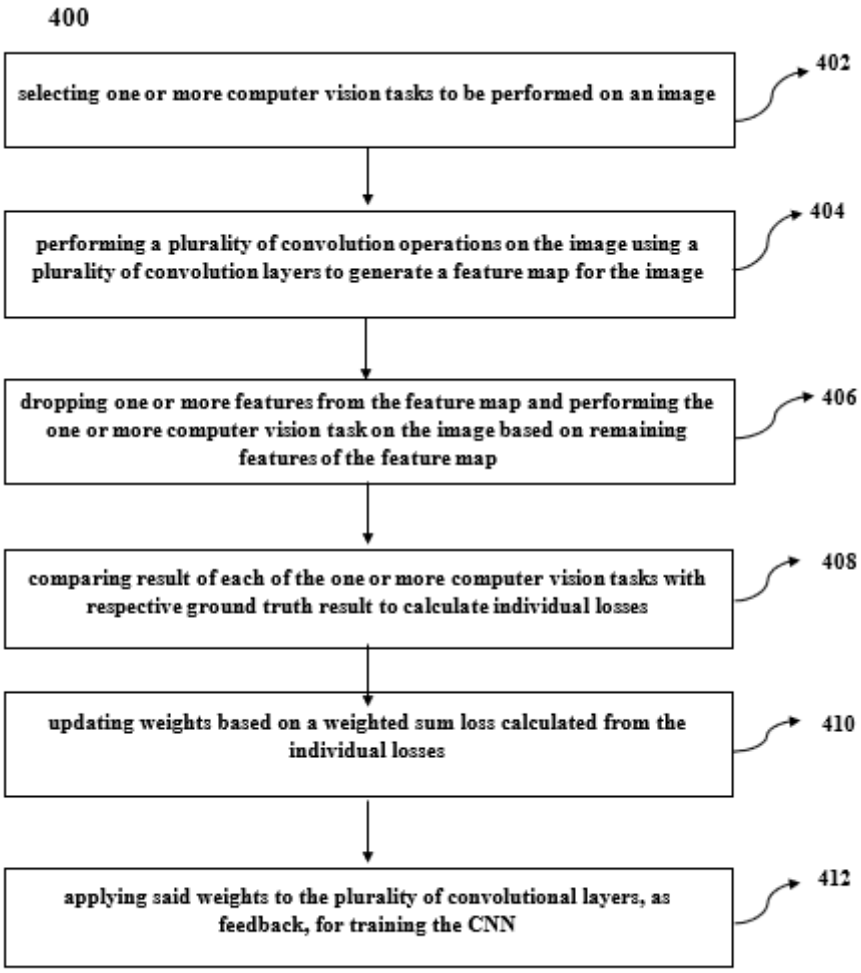

FIG. 4

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

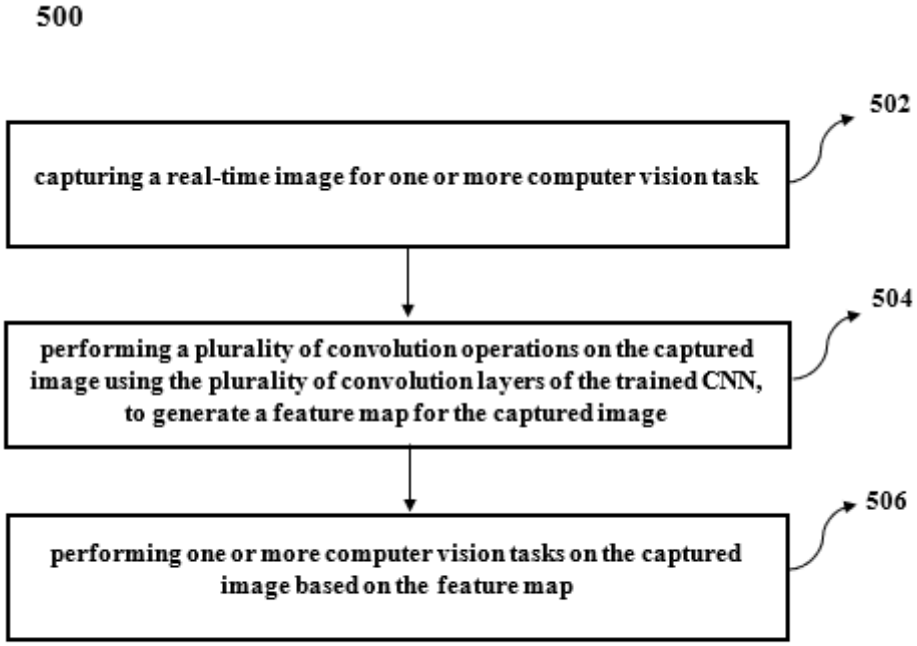

FIG. 5

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

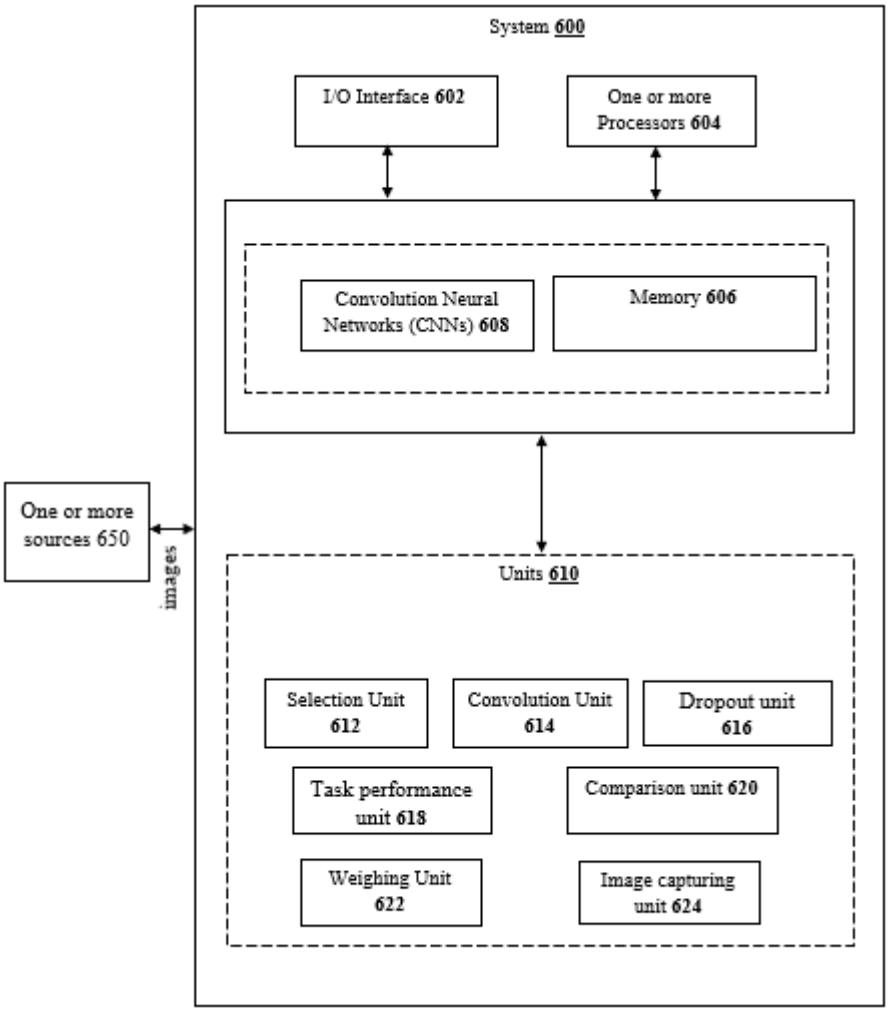

Fig. 6

-- Digitally Signed--
Robin Koshy Varghese
(INPA No: 3705)
Head, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.