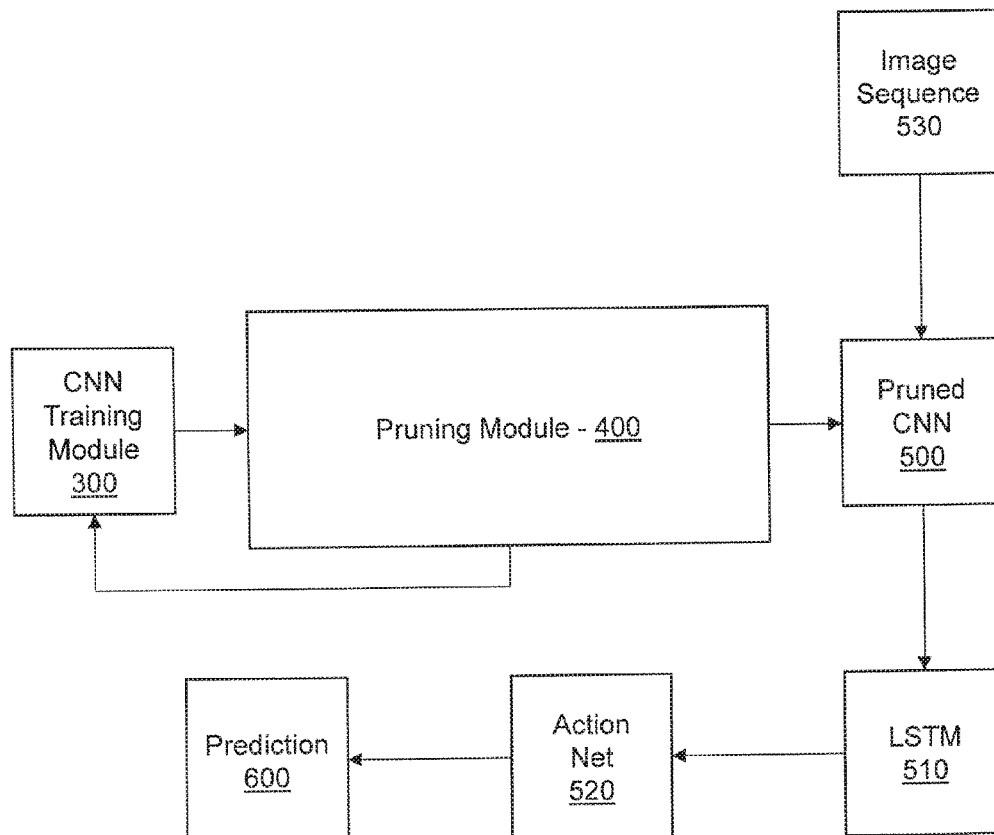


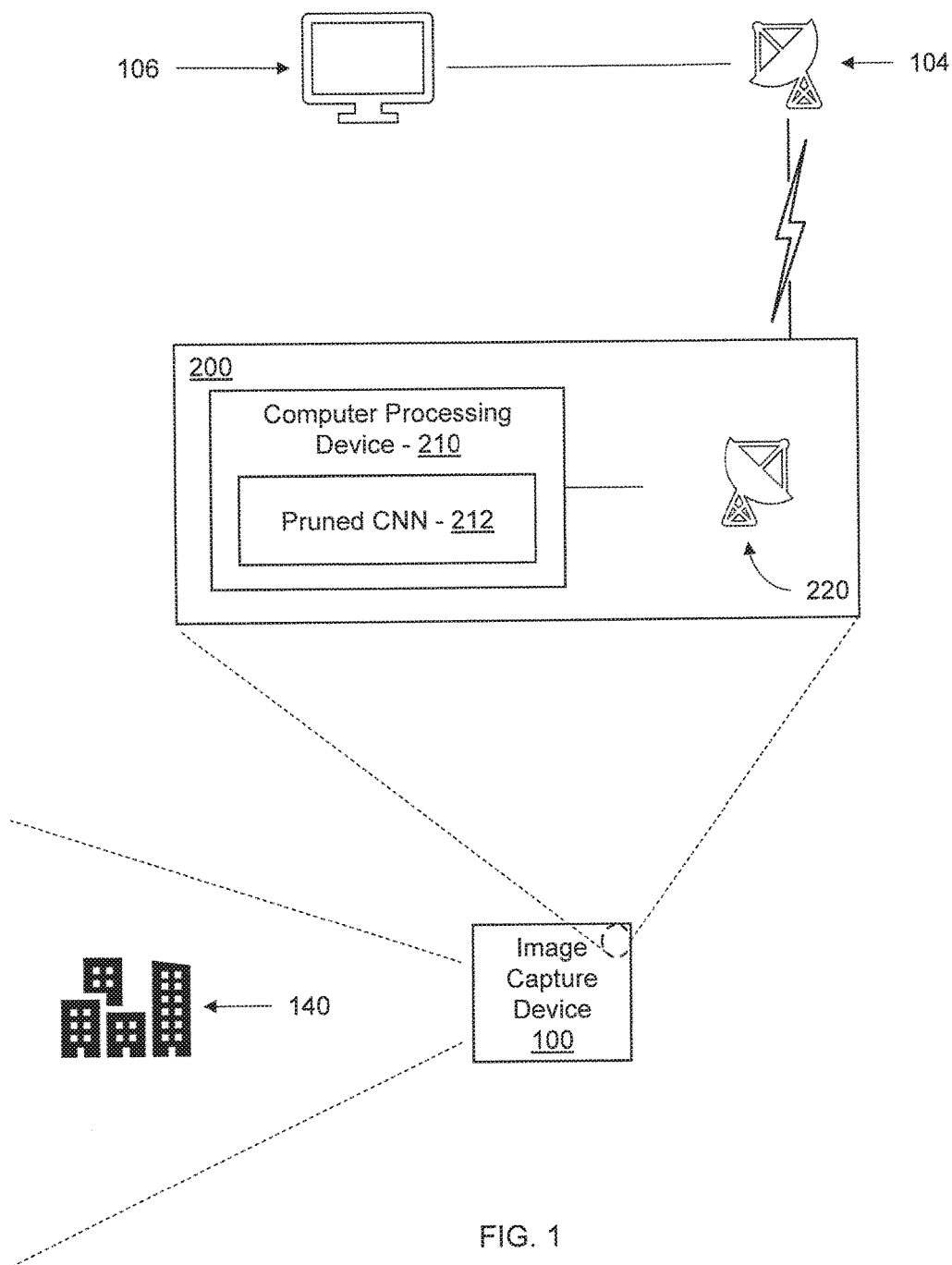


US 20180336468A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2018/0336468 A1**
(43) **Pub. Date: Nov. 22, 2018**(54) **PRUNING FILTERS FOR EFFICIENT
CONVOLUTIONAL NEURAL NETWORKS
FOR IMAGE RECOGNITION IN
SURVEILLANCE APPLICATIONS**(52) **U.S. Cl.**
CPC *G06N 3/082* (2013.01); *G06K 9/00771*
(2013.01); *G06N 5/046* (2013.01)(71) Applicant: **NEC Laboratories America, Inc.,**
Princeton, NJ (US)(57) **ABSTRACT**(72) Inventors: **Asim Kadav**, Jersey City, NJ (US);
Igor Durdanovic, Lawrenceville, NJ
(US); **Hans Peter Graf**, South Amboy,
NJ (US)(21) Appl. No.: **15/979,500**(22) Filed: **May 15, 2018****Related U.S. Application Data**(60) Provisional application No. 62/506,657, filed on May
16, 2017.**Publication Classification**(51) **Int. Cl.**
G06N 3/08 (2006.01)
G06N 5/04 (2006.01)
G06K 9/00 (2006.01)

Systems and methods for pruning a convolutional neural network (CNN) for surveillance with image recognition are described, including extracting convolutional layers from a trained CNN, each convolutional layer including a kernel matrix having at least one filter formed in a corresponding output channel of the kernel matrix, and a feature map set having a feature map corresponding to each filter. An absolute kernel weight is determined for each kernel and summed across each filter to determine a magnitude of each filter. The magnitude of each filter is compared with a threshold and removed if it is below the threshold. A feature map corresponding to each of the removed filters is removed to prune the CNN of filters. The CNN is retrained to generate a pruned CNN having fewer convolutional layers to efficiently recognize and predict conditions in an environment being surveilled.





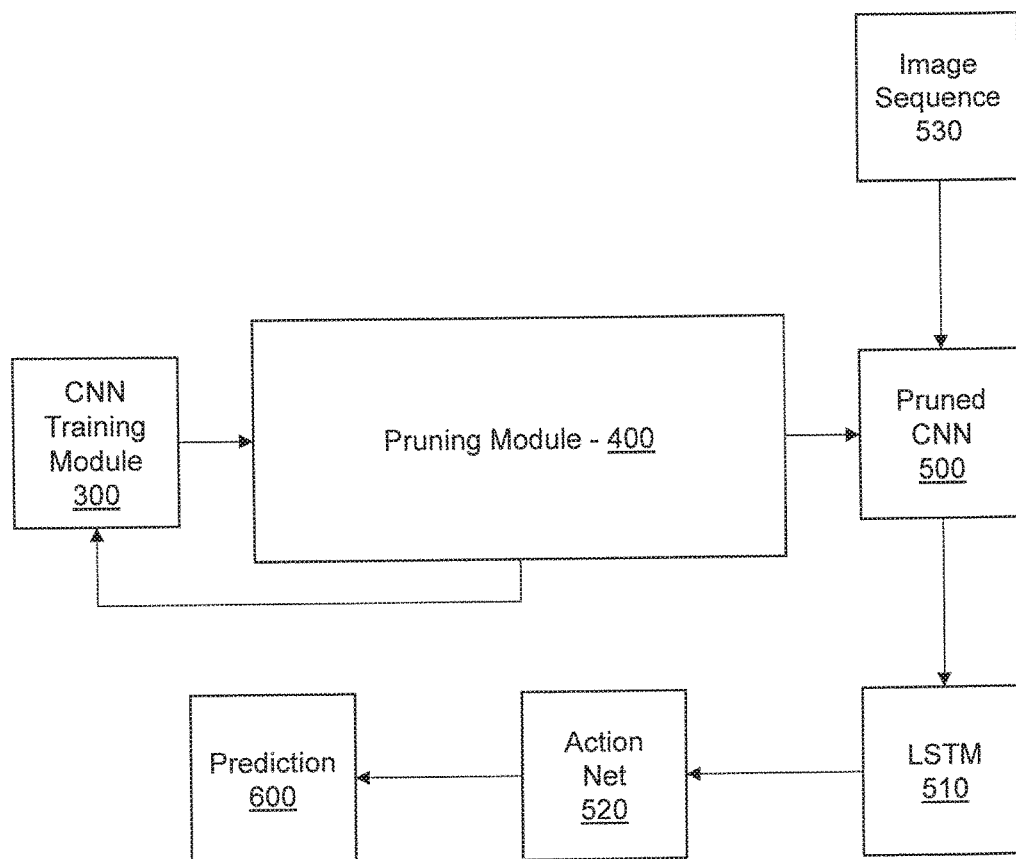


FIG. 2

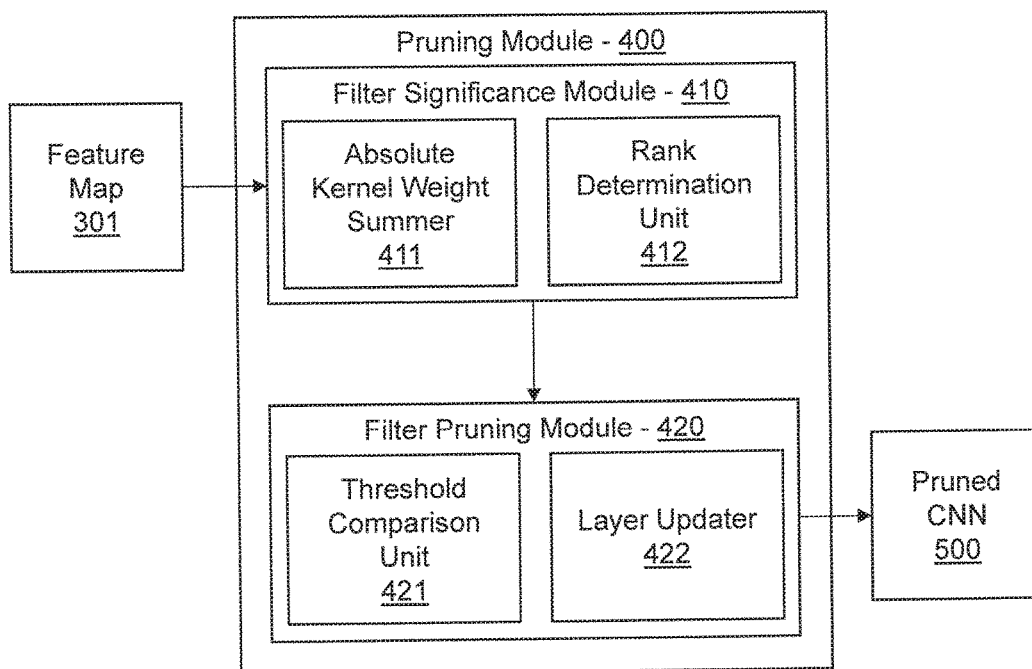


FIG. 3

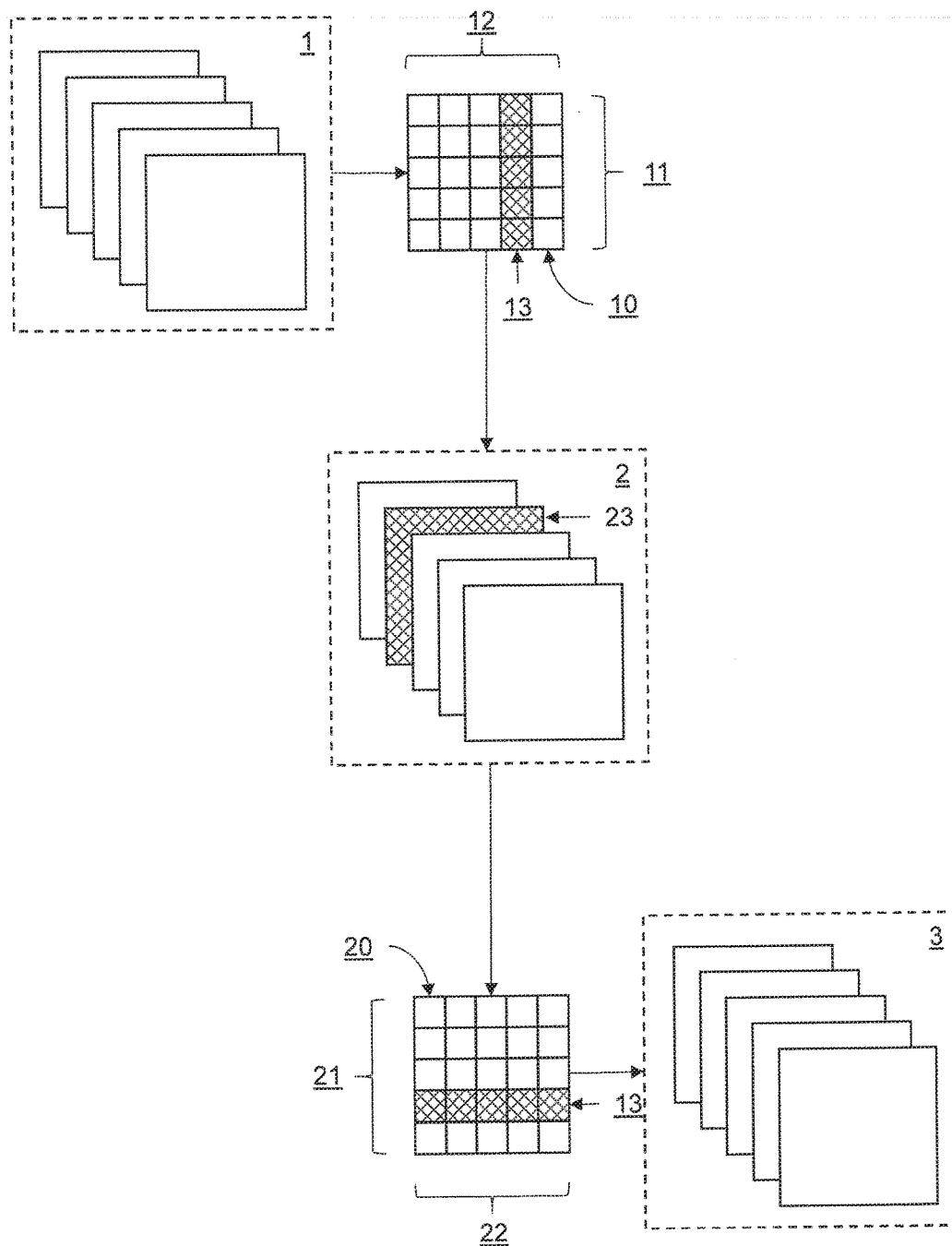


FIG. 4

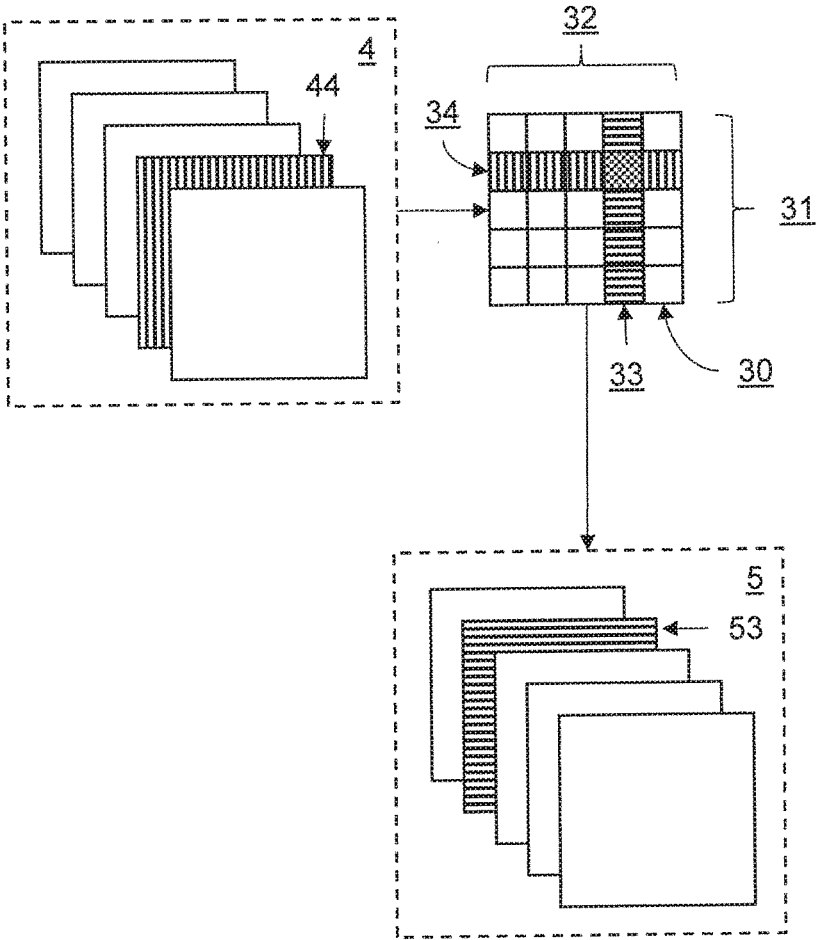


FIG. 5

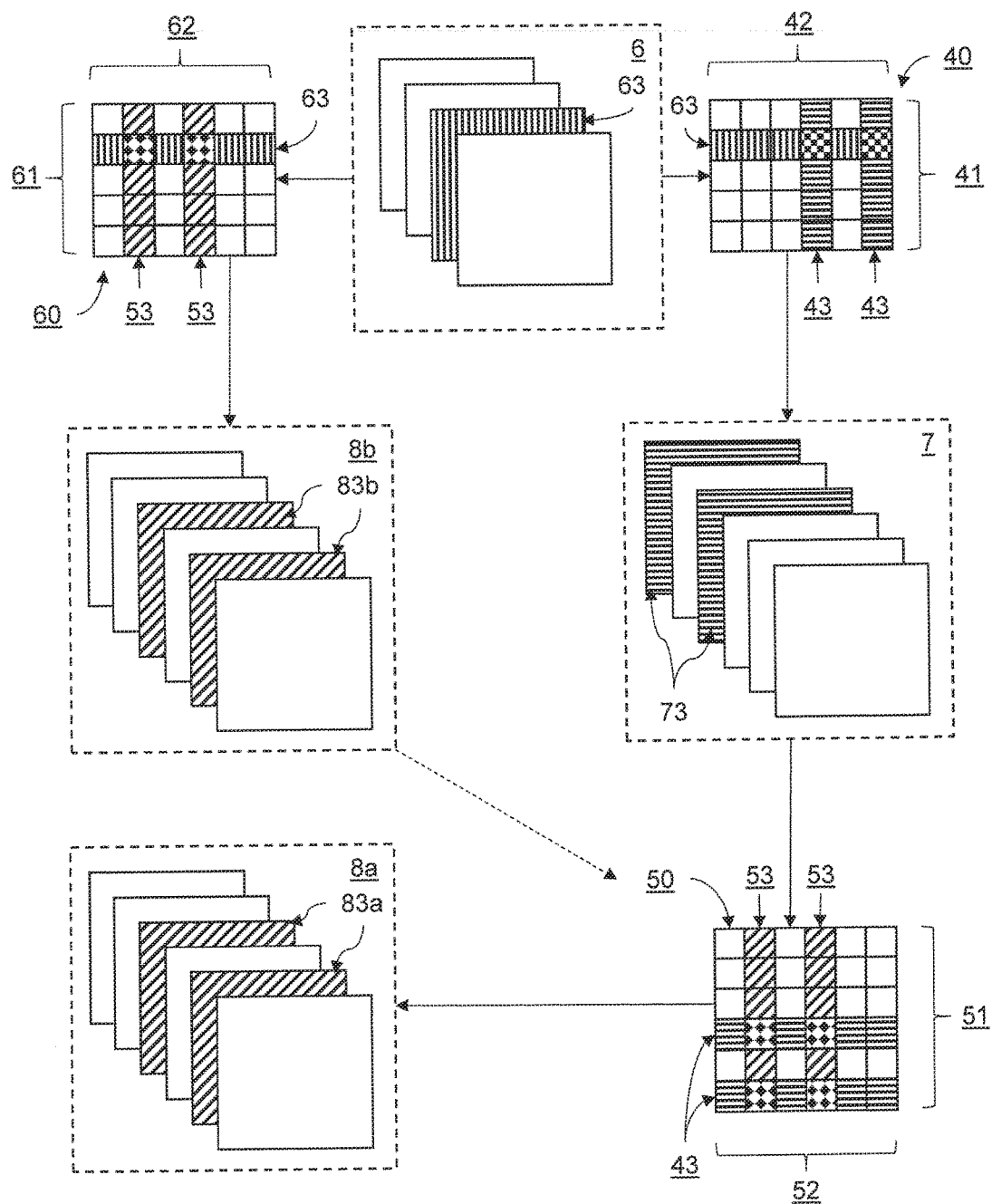


FIG. 6

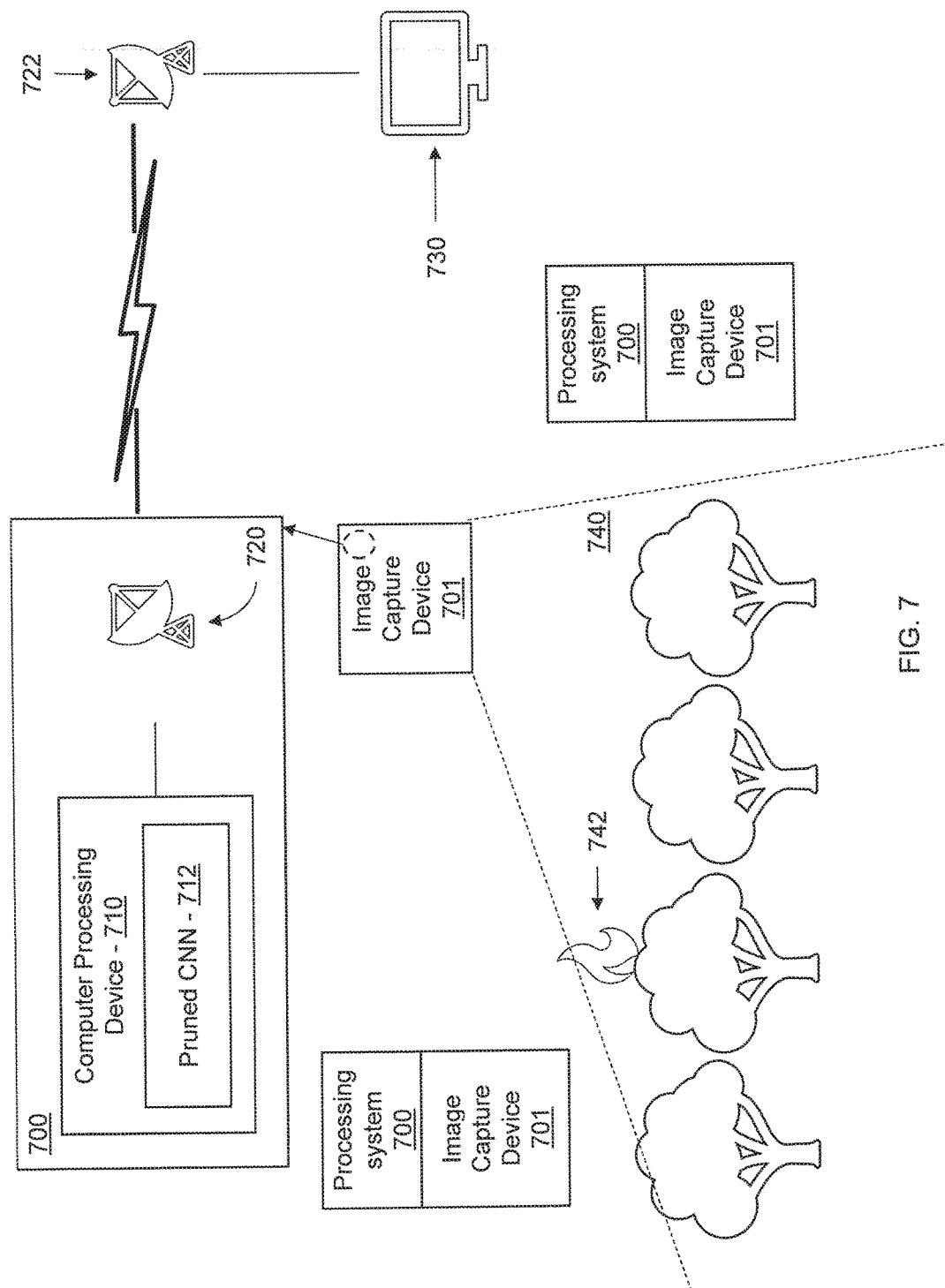


FIG. 7

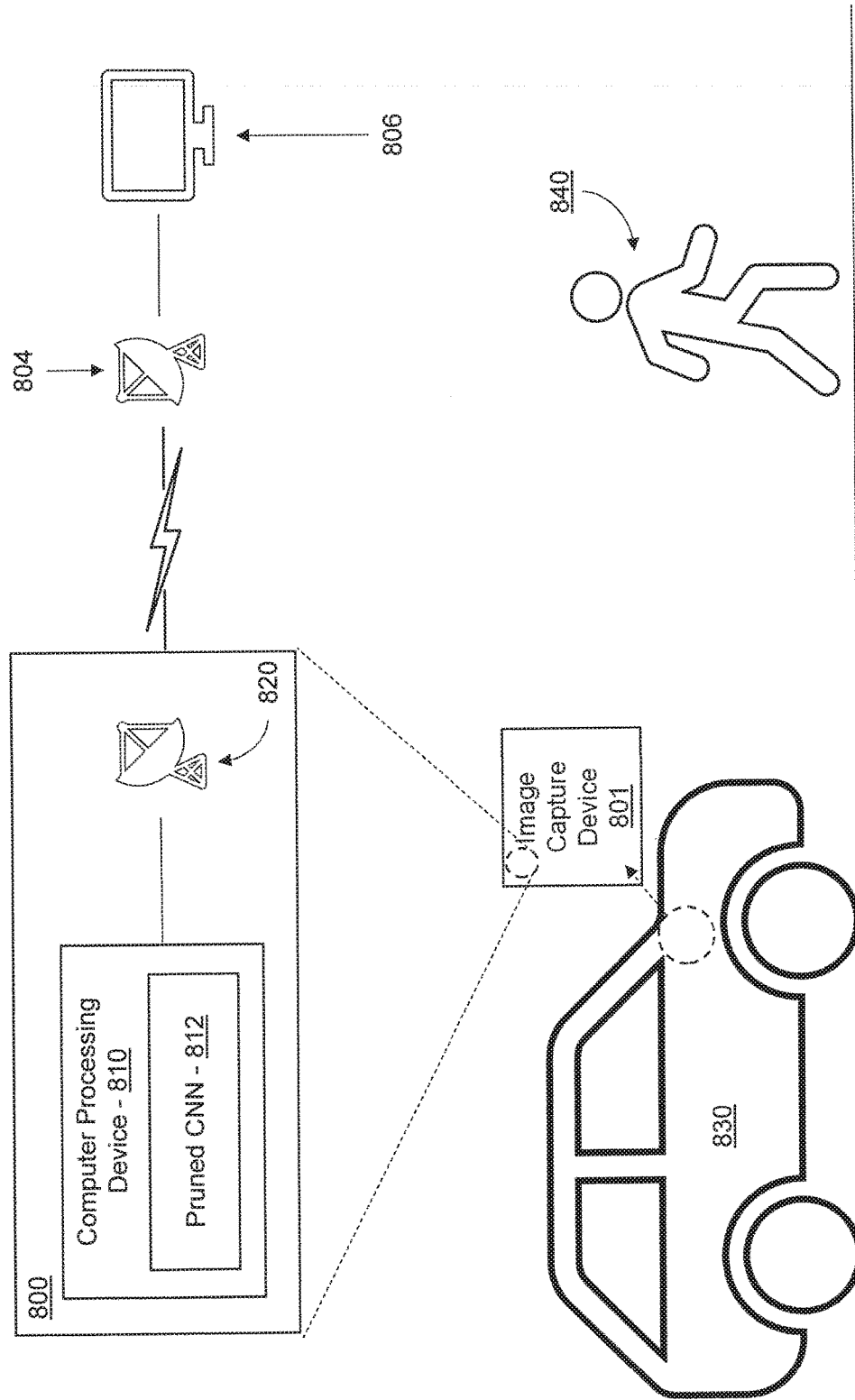


FIG. 8

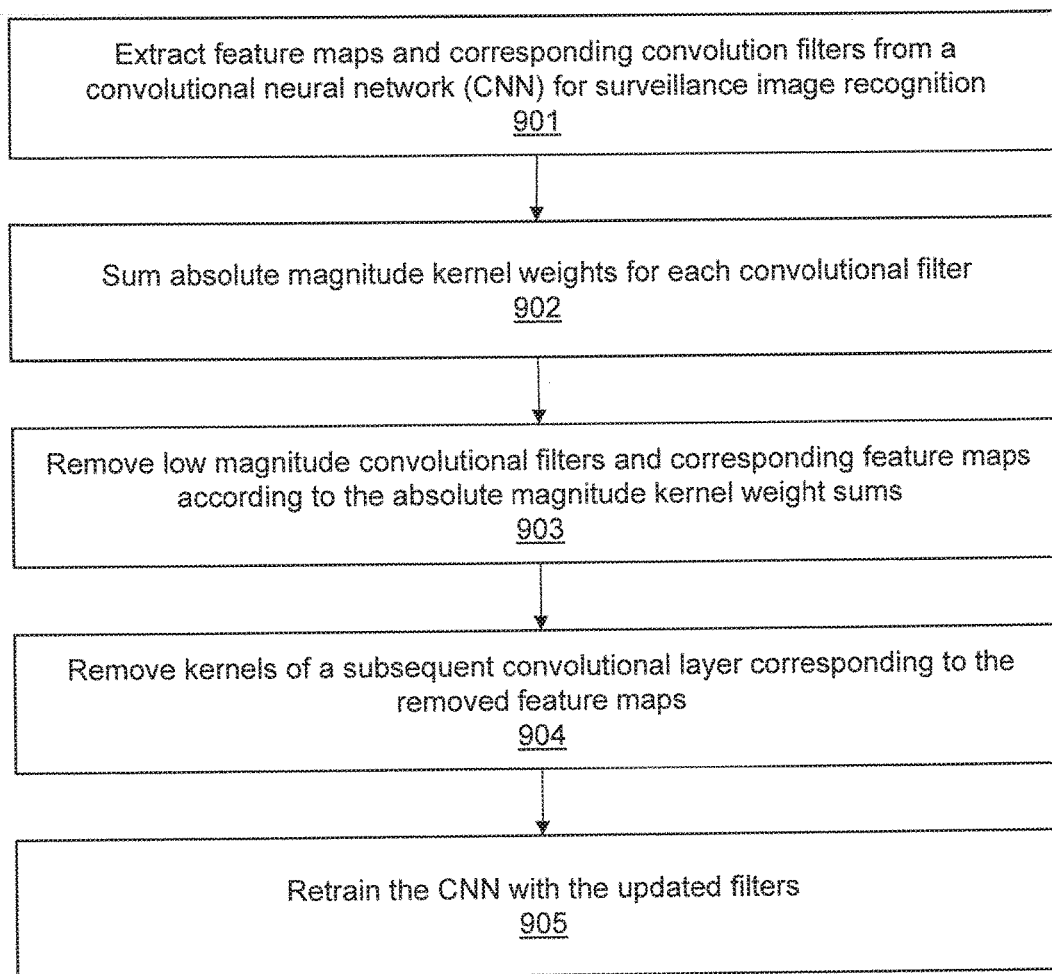


FIG. 9

**PRUNING FILTERS FOR EFFICIENT
CONVOLUTIONAL NEURAL NETWORKS
FOR IMAGE RECOGNITION IN
SURVEILLANCE APPLICATIONS**

RELATED APPLICATION INFORMATION

[0001] This application claims priority to 62/506,657, filed on May 16, 2017, incorporated herein by reference in its entirety. This application is related to an application entitled “PRUNING FILTERS FOR EFFICIENT CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE RECOGNITION OF ENVIRONMENTAL HAZARDS”, having attorney docket number 16085B, and an application entitled “PRUNING FILTERS FOR EFFICIENT CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE RECOGNITION IN VEHICLES”, having attorney docket number 16085C, and which are incorporated by reference herein in their entirety.

BACKGROUND

Technical Field

[0002] The present invention relates to image recognition with neural networks and more particularly image recognition filter pruning for efficient convolutional neural networks for surveillance applications.

Description of the Related Art

[0003] Convolutional neural networks (CNNs) can be used to provide image recognition. As image recognition efforts have become more sophisticated, so have CNNs for image recognition, using deeper and deeper networks with greater parameters and convolutions. However, this trend also results in a greater need of the CNN for computational and power resources. Thus, image recognition with CNNs is impractical, and indeed, in some instances, impossible in embedded and mobile application. Even application that have the power and computational resources for accurate CNNs would benefit from more efficient image recognition. Simply compressing or pruning the weights of layers of a neural network would not adequately reduce the costs of a deep neural network.

SUMMARY

[0004] According to an aspect of the present principles, a method is provided for pruning a convolutional neural network (CNN) for surveillance with image recognition is described. The method includes extracting at least one convolutional layer from a trained CNN, each convolutional layer including a kernel matrix having at least one filter formed in a corresponding output channel of the kernel matrix, and feature map set having a feature map corresponding to each of the at least one filter. An absolute kernel weight is determined for each kernel in the kernel matrix. The absolute kernel weights of each kernel in each of the at least one filter are summed to determine a magnitude of each filter. The magnitude of each filter is compared with a threshold and removing one or more filters that are below the threshold. A feature map corresponding to each of the removed filters is removed to prune the CNN of filters. The CNN is retrained upon pruning the removed filters to

generate a pruned CNN having fewer convolutional layers to efficiently recognize and predict conditions in an environment being surveilled.

[0005] According to an aspect of the present principles, a non-transitory computer readable storage medium comprising a computer readable program for surveillance with image recognition using a pruned convolutional neural network (CNN) is described. The computer readable program when executed on a computer causes the computer to perform the steps including extracting at least one convolutional layer from a trained CNN, each convolutional layer including a kernel matrix having at least one filter formed in a corresponding output channel of the kernel matrix, and feature map set having a feature map corresponding to each of the at least one filter. An absolute kernel weight is determined for each kernel in the kernel matrix. The absolute kernel weights of each kernel in each of the at least one filter are summed to determine a magnitude of each filter. The magnitude of each filter is compared with a threshold and removing one or more filters that are below the threshold. A feature map corresponding to each of the removed filters is removed to prune the CNN of filters. The CNN is retrained upon pruning the removed filters to generate a pruned CNN having fewer convolutional layers to efficiently recognize and predict conditions in an environment being surveilled.

[0006] According to another aspect of the present principles, a system is provided for image recognition system for surveillance. The system includes an image capture device for capturing images of an environment to be surveilled. An image recognition system is included in an embedded computing device included in the image capture device configured to perform image recognition with a pruned CNN. The image recognition system includes an absolute kernel weight summer configured to determining an absolute kernel weight for each kernel in the kernel matrix and sum the absolute kernel weights of each kernel in a filter corresponding to each of at least one output channel of a kernel matrix to determine a magnitude of each filter, each filter corresponding to a feature map. The image recognitions system further includes a threshold comparison unit configured to comparing the magnitude of each filter with a threshold and removing one or more filters that are below the threshold and a layer updater configured to removing a feature map corresponding to each of the removed filters to prune the CNN of filters and generate the pruned CNN. A long short-term memory network (LSTM) is included for predicting feature actions. An action network is included for generating class probabilities of feature actions. A notification device is included for notifying a user of the class probabilities.

[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 high-level system/method for surveillance with image recognition using a pruned convolution neural network (CNN), in accordance with the present principles;

[0010] FIG. 2 is a block/flow diagram illustrating a system/method for image recognition using a pruned convolution neural network (CNN), in accordance with the present principles;

[0011] FIG. 3 is a block/flow diagram illustrating a system/method for pruning filters of a convolution neural network (CNN), in accordance with the present principles;

[0012] FIG. 4 is a block/flow diagram illustrating a system/method for pruning filters of a simple convolution neural network (CNN), in accordance with the present principles;

[0013] FIG. 5 is a block/flow diagram illustrating a system/method for pruning intermediate filters of a simple convolution neural network (CNN), in accordance with the present principles;

[0014] FIG. 6 is a block/flow diagram illustrating a system/method for pruning intermediate filters of a residual convolution neural network (CNN), in accordance with the present principles;

[0015] FIG. 7 is a block/flow diagram illustrating a high-level system/method for surveillance of forest fires with image recognition using a pruned convolution neural network (CNN), in accordance with the present principles;

[0016] FIG. 8 is a block/flow diagram illustrating a high-level system/method for surveillance for vehicles with image recognition using a pruned convolution neural network (CNN), in accordance with the present principles; and

[0017] FIG. 9 is a flow diagram illustrating a system/method for image recognition using a pruned convolution neural network (CNN), in accordance with the present principles.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0018] In accordance with the present principles, systems and methods are provided for a convolutional neural network (CNN) trained with pruned filters for image recognition in surveillance applications.

[0019] In one embodiment, the number of filters in a CNN is reduced by pruning. This pruning is accomplished by training a CNN for image recognition in a surveillance application. Once trained, the filters of the CNN can be assessed by determining the weights of each filter. By removing the filters that have small weights, the filters that have little contribution to accuracy can be removed, and thus pruned.

[0020] Once the filters have been pruned, the CNN can be retrained until it reaches its original level of accuracy. Thus, fewer filters are employed in a CNN that is equally accurate. By removing filters, the number of convolution operations and reduced, thus reducing computation costs, including computer resource requirements as well as power requirements. This pruning process also avoids the need to maintain sparse data structures or sparse convolution libraries because the filters having lower contributions are completely removed. As a result, the pruned CNN can be made efficient enough to be employed in embedded devices and mobile devices such, e.g., digital cameras and camcorders, personal computers, tablets, smartphones, vehicles, drones, satellites, among others. Predictions may be made of future situations and actions according to the recognized images. Thus, a surveillance system employing the pruned CNN can leverage the more efficient image recognition to achieve situation

predictions early and more quickly so that more effective and better-informed actions may be proactively taken.

[0021] Embodiments described herein may be entirely hardware, entirely software or including both hardware and software elements. In a preferred embodiment, the present invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0022] Embodiments may include a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. A computer-usable or computer readable medium may include any apparatus that stores, communicates, propagates, or transports the program for use by or in connection with the instruction execution system, apparatus, or device. The medium can be magnetic, optical, electronic, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. The medium may include a computer-readable storage medium such as a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk, etc.

[0023] Each computer program may be tangibly stored in a machine-readable storage media or device (e.g., program memory or magnetic disk) readable by a general or special purpose programmable computer, for configuring and controlling operation of a computer when the storage media or device is read by the computer to perform the procedures described herein. The inventive system may also be considered to be embodied in a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer to operate in a specific and predefined manner to perform the functions described herein.

[0024] A data processing system suitable for storing and/or executing program code may include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code to reduce the number of times code is retrieved from bulk storage during execution. Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) may be coupled to the system either directly or through intervening I/O controllers.

[0025] Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

[0026] Referring now in detail to the figures in which like numerals represent the same or similar elements and initially to FIG. 1, a high-level system/method for surveillance with image recognition using a pruned convolution neural network (CNN) is illustratively depicted in accordance with one embodiment of the present principles.

[0027] In one embodiment, a system is contemplated for providing surveillance for an area of interest 140. The area of interest 140 may be, e.g., the interior or exterior of a building (for example, a shopping mall, airport, office building, etc.), a parking lot, the interior or exterior of a house, or

any other public or private place for which surveillance may be desired. According to aspects of the present invention, surveillance may include image recognition to facilitate the recognition and response to potentially dangerous or hazardous situations, such as, e.g., fires, floods, criminal activity, or any other natural or man-made condition in the area of interest **140**.

[0028] Surveillance may be performed by an image capture device **100**. The image capture device **100** is used to capture an image or sequence of images of the area of interest **140** such that the image or sequence of images may be analyzed to recognize or predict a hazardous situation. Accordingly, the image capture device **100** may include, e.g., a still or video camera, or any device incorporating a suitable sensor for capturing images (for example, e.g., a device including a charge-coupled device (CCD), a photodiode, a complementary metal oxide semiconductor (CMOS) sensor, infrared sensor, laser distance and ranging (LIDAR) sensor, among others).

[0029] The image capture device **100** may include a processing system **200** for performing the analysis, including image recognition. The processing system **200** may have an internal component to the image capture device **100**, or may be external and in communication to the image capture device **100**. Accordingly, the processing system **200** may include, e.g., a system-on-chip (SOC), a computer processing unit (CPU), a graphical processing unit (GPU), a computer system, a network, or any other processing system.

[0030] The processing system **200** may include a computer processing device **210** for processing the image or sequence of images provided by the image capture device **100**. Accordingly, the computer processing device **210** can include, e.g., a CPU, a GPU, or other processing device or combination of processing devices. According to aspects of an embodiment of the present invention, the processing system **200** is an embedded processing system including, e.g., a SOC, or a mobile device such as, e.g., a smartphone.

[0031] Such systems have strict power and resource constraints because they do not benefit from grid power or large packaging then provide a large volume of power, storage or processing power. Therefore, to perform the image recognition, the computer processing device **210** includes a pruned CNN **212**. The pruned CNN **212** reduces the complexity of the CNN for image recognition by pruning particular portions of the CNN that have a relatively insignificant impact on the accuracy of the image recognition. Thus, pruning a CNN can reduce the resource requirements of the CNN by reducing complexity, while having minimal impact on the accuracy of the CNN. As a result, the pruned CNN **212** may be, e.g., stored in volatile memory, such as, e.g., dynamic random access memory (DRAM), processor cache, random access memory (RAM), or a storage device, among other possibilities. Accordingly, the pruned CNN **212** may be stored locally to an embedded or mobile processing system and provide accurate image recognition results.

[0032] It has been found that convolution operations contribute significantly towards overall computation. Indeed, the convolution operations themselves can contribute to up to about 90.9% of overall computation effort. Therefore, reducing the number of convolution operations through pruning will cause a corresponding reduction in computing resources demanded by the operation of the CNN, such as, e.g., electrical power, processing time, memory usage, storage usage, and other resources.

[0033] Thus, according to aspects of the present invention, the pruning of the CNN can take the form, e.g., of pruning filters from convolution layers of the CNN. According to aspects of one possible embodiment, a fully trained CNN is tested at each layer to locate the filters with small weights. Small weights may be determined, e.g., based on their relation to other filter weights (e.g., a number of the smallest weighted filters), or by being below a certain threshold value, or by another suitable standard. Alternatively, or in addition, sensitivity to pruning may be a determining factor for whether a filter is pruned. Sensitivity may be assessed by, e.g., removing a filter and testing the effect of the removal on accuracy to determine if it exceeds a threshold value at which a filter is deemed too sensitive pruning, and thus is replaced into the kernel matrix, or by comparing the weights of each kernel in the filter to a threshold magnitude, and if there are too few kernel weight weights with a magnitude below the threshold then the filter is deemed sensitive to pruning and should not be removed.

[0034] The filters with the small weights or that are not sensitive to pruning are then removed from the CNN. Thus, entire convolutional operations corresponding to the removed filters may be eliminated from the CNN. After removal, the CNN may then be retrained back to its original accuracy without the removed filters. Accordingly, the pruned CNN **212** has reduced resource requirements and can, therefore, be more effectively implemented in the processing system **200**. The pruning and retraining of the CNN may take place, e.g., on a separate computing system before being implemented in the processing system **200**, or it may be performed by the processing system **200** itself.

[0035] The image recognition results from the pruned CNN **212** can be combined with action prediction to predict changes to situations. Thus, the processing system **200** analyzes the recognized images from the pruned CNN **212** to predict future movement and actions of features in the image or images. Thus, dangerous and hazardous situations can be predicted and proactively addressed. The action prediction can be performed by the pruned CNN **212**, or, e.g., by additional features such as, e.g., recurrent neural networks (RNN) including long short-term memory, among other prediction solutions.

[0036] The results, including the image recognition results and any predictions, communicated from the computer processing **210** to a transmitter **220**. The transmitter **220** may be included in the processing system **200**, or it may be separate from the processing system **200**. The transmitter **220** communicates the image recognition results to a remote location or device via a receiver **104** at that location or device. The communication may be wired or wireless, or by any suitable means to provide the image recognition results.

[0037] Upon receipt of the image recognition results, the receiver **104** communicates the results to a notification system **106**. The notification system **106** provides a user with a notification of the recognized images, such as, an alert of a hazardous situation (for example, an alarm, flashing lights, visual message on a display, or auditory message from a speaker, among others), a display of the recognized images including, e.g., present or predicted labelled images, object lists, actions, or conditions. Accordingly, the notification system **106** may take the form of, e.g., a display, a speaker, an alarm, or combinations thereof. Thus, a user may be notified of the features that have been recognized in an image by the pruned CNN **212**, and act accordingly.

[0038] Referring now to FIG. 2, a system/method for image recognition using a pruned convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0039] According aspects of an embodiment of the present invention, image recognition for surveillance may be performed to predict future actions of people and things in an environment. The image recognition is performed on an input image sequence 530. The image sequence 530 is a series of still or video images of an environment being surveilled.

[0040] The image sequence 530 is received by a pruned CNN 500. The pruned CNN 500 may be implemented in a computing device that has few resources to expend on convolutions. As discussed above, this computing device can include, e.g., embedded systems or mobile devices, among others. Therefore, power and computing resources are limited. However, using one of these computing devices allows for the use of surveillance equipment that is mobile, fast, and cheap. For example, the surveillance equipment could include, e.g., a smartphone, a drone, a smart doorbell system, a digital camera or camcorder, or any other suitable device. However, to facilitate image recognition on-board the surveillance equipment or by a similarly mobile, fast, cheap device in communication with the surveillance equipment permits image recognition on-site in locations where access to high computing power and large power sources is not feasible. Therefore, a pruned CNN 500 is implemented that has fewer computation requirements.

[0041] The pruned CNN 500 may be generated from a CNN trained by a CNN training module 300. The CNN training module 300 may be external to the surveillance equipment, such that upon training and pruning, the pruned CNN 500 is transferred to the surveillance equipment. However, the CNN training module 300 may alternatively be on-board the surveillance equipment.

[0042] The CNN training module 300 trains a CNN that can include a suitable CNN for image recognition, such as, e.g., a simple CNN including VGG-16, or a more complicated CNN including a residual CNN such as RESNET. The CNN may be trained using any suitable dataset, such as, e.g., CIFAR-10. The CNN training module 300 uses that dataset to input each image, and for each image, apply filters and generate intermediate feature maps through the depth of the CNN, testing the final feature maps and updating weights of the filters. This process may be performed any suitable number of times until the CNN is trained.

[0043] The trained CNN is then provided to a pruning module 400. The pruning module 400 prunes the CNN such that fewer convolutional operations need to be performed to generate a final feature map of the image recognition. Accordingly, the pruning module 400 can, for example, e.g., remove filters from the CNN that are not significant to the image recognition task. A filter can be determined to not be significant because it is relatively rarely activated during image recognition. Thus, a filter that does not meet a threshold that represents the frequency of activation should be pruned. The threshold could be a weight value. In such a case, the weights of the filter are assessed and compared to the threshold. If the threshold is not met, then the filter is pruned from the CNN to reduce convolutional operations. Alternatively, the threshold could be based on a number for filters. In this case, the weights of the filters are assessed and the filters can be ranked according to their weights. A

threshold number of the smallest filters can then be pruned from the CNN to reduce convolutional operations. Assessment of the weights can include any suitable assessment, such as, e.g., determining an average or a median weight for a filter, determining a sum of the weights, or determining an absolute sum of the weights for a filter, among other techniques for assessment.

[0044] Accordingly, filters are removed from the CNN, reducing the convolutional operations of the CNN, and thus reducing computational requirements. Upon pruning, the CNN may be returned the CNN training module 300 to be retrained without the pruned filters. This retraining may be performed after each filter is removed, or it may be performed after pruning filters through the depth of the CNN. The CNN may be retrained until it reaches the original level of accuracy of the trained CNN prior to pruning.

[0045] Upon pruning the CNN, the pruning module 400 provides the pruned CNN 500 to the surveillance equipment for image recognition, as discussed above. The pruned CNN 500 receives the image sequence 530 and performs image recognition to recognize objects and conditions present in each image. The image recognition results may take the form of image feature vectors, each image feature vector containing information about objects and conditions in a given image. Accordingly, in one embodiment, each image feature vector can include, e.g., 2048 dimensions for robust feature determination corresponding to image recognition.

[0046] The image recognition can include predictions about the objects and conditions, such as future movement. These predictions assist a user with surveillance by predicting possible threats, hazards or dangers. Therefore, the image feature vectors from the pruned CNN 500 can be provided to, e.g., a recurrent neural network such as, e.g., a long short-term memory network (LSTM) 510. The LSTM 510 will have been trained to use the image feature vectors to generate an action feature vector that can be, e.g., smaller than the image feature vector. In one embodiment, the action feature vector generated by the LSTM 510 can include, e.g., a 1024-dimension action feature vector. The LSTM 510 generates, with the action feature vector, predictions of future actions of the objects and conditions in the image sequence 530.

[0047] In one embodiment, action feature vector can then be processed to generate probabilities of predicted actions. Accordingly, an action net 520 is employed to receive the action feature vector. The action net 520 will have been trained to use the action feature vector to generate n class output probabilities that form the probabilities of predicted actions to generate a prediction 600 corresponding to the surveillance.

[0048] Referring now to FIG. 3, a system/method for pruning filters of a convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0049] According to aspects of an embodiment of the present invention, the pruning module 400 prunes filters from a trained CNN based on, e.g., a sum of absolute kernel weights in the filter. The filters are pruned by first inputting the trained CNN 301. The trained CNN 301 includes initial, intermediate, and final feature maps with associated trained filters and kernel matrices. In training, weights of the kernel matrices are determined by inputting training images, apply-

ing the existing filters with kernel matrices, and updated the weights of the kernel matrices based on an error of the output.

[0050] Pruning unnecessary filters facilitates reducing convolutional operations of the trained CNN 301. Therefore, unnecessary filters are determined by a filter significance module 411 of the pruning module 400.

[0051] The filter significance module 411 includes an absolute kernel weight summer 411. The absolute kernel weight summer 411 calculates the absolute kernel weights of a filter, and sums the absolute kernel weights. The sum of the absolute kernel weights is representative of the magnitude of the filter to which it applies. Therefore, a lower absolute kernel weight sum will indicate that a filter has less effect on the image recognition process, and is thus activated less frequently. Accordingly, the lower the absolute kernel weight sum of a filter, the lower the significance of that filter.

[0052] Furthermore, sensitivity to pruning can be a determining factor for whether a filter is pruned. Sensitivity may be assessed by, e.g., removing a filter and testing the effect of the removal on accuracy to determine if it exceeds a threshold value at which a filter is deemed too sensitive pruning, and thus is replaced into the kernel matrix, or by comparing the weights of each kernel in the filter to a threshold magnitude, and if there are too few kernel weight weights with a magnitude below the threshold then the filter is deemed sensitive to pruning and should not be removed. Other methods of determining sensitivity to pruning are also contemplated.

[0053] According to aspects of the present invention, based on the absolute kernel weight sums calculated by the absolute kernel weight summer 411, the filters of a given set of feature maps can be ranked by the rank determination unit 412. The rank of a given filter may be determined by the magnitude of the absolute kernel weight sum pertaining to that filter, with the rank forming, e.g., a spectrum of filter magnitudes from least to greatest. However, other ranking methodologies are contemplated.

[0054] The ranks filters may then be analyzed by the filter pruning module 420. The filter pruning module 420 includes a threshold comparison unit 421. This threshold comparison unit 421 is configured to compare the absolute kernel weight sums for each of the filters with a threshold. The threshold can be predetermined or can change according to a distribution of absolute kernel weight sums. For example, the threshold can include a threshold absolute kernel weight sum. In this embodiment, the absolute kernel weight sums of each filter are compared with the threshold value, and if the absolute kernel weight sum of a given filter is below the threshold value, then the filter is removed. Alternatively, the threshold can be a threshold number of filters. Thus, the threshold comparison unit 421 may compare the number of filters that exist in the trained CNN 301 at a given convolutional layer with a threshold number of filters. A difference, m , may then be taken between the threshold value and the number of filters, with the difference, m , indicating the number of filters to be removed. Then, m of the smallest filters are removed. However, this threshold value may instead indicate a set number of the smallest filter to be removed. For example, rather than a difference, m may be a predetermined value of the smallest filters to be removed from the convolutional layer.

[0055] Once filters of a given convolutional layer are removed, the CNN can be updated to reflect the removal.

Thus, a layer updater 422 will update the trained CNN 301 by removing the kernels for the subsequent convolutional layer that corresponds to the removed filter. The kernel matrices for the given and the subsequent convolutional layers can then be reconstructed to reflect the removed filters and removed kernels.

[0056] Each convolutional layer of the trained CNN 301 will be processed by the pruning module 400 as described above. The CNN can be retrained iteratively after each layer is pruned, or all of the layers may be pruned before retraining the CNN. However, in either case, upon pruning all of the layers, the pruning module 400 will output a pruned CNN 500 to be used for image recognition. The pruned CNN 500 will be significantly less complex than the original trained CNN 301, while maintaining substantially the same level of accuracy because only the insignificant filters for the image recognition task are removed. Thus, the pruned CNN 500 can be incorporated, as discussed above, into a system have low computational or power resources, such as, e.g., embedded or mobile systems. Additionally, the pruned CNN 500 will also be faster than the original trained CNN 301 due to the reduction in convolutional operations. Thus, even if used in a system having high resources, the pruned CNN 500 is still beneficial because the pruned CNN 500 is more efficient and can therefore perform the surveillance image recognition task more quickly, thus having benefits even in applications where resource constraints are not a concern.

[0057] Referring now to FIG. 4, a system/method for pruning filters of a simple convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0058] According to an embodiment of the present invention, filter 13 pruned from a kernel matrix 10 is illustrated. Pruning includes a set of input feature maps 1 and intermediate and/or output feature maps 2 and 3. The input feature maps 1 have filters applied with kernel matrix 10 to generate feature maps 2. Similarly, feature maps 2 have filters applied with kernel matrix 20 to generate feature maps 3.

[0059] A feature map 13 may be pruned from feature maps 2 using the absolute kernel weight sums from kernel matrix 10, where each box in the kernel matrix 10 corresponds to a kernel for a filter. A vertical axis of the kernel matrix 10 is the input channels 11 while the horizontal axis is the output channels 12. Thus, the columns of kernel matrix 10 form a 3D filter for generating the feature maps 2.

[0060] To determine the magnitude, and thus significance, of a filter, absolute kernel magnitudes of each kernel in a given column are summed. Each filter can then be compared to a threshold, as discussed above. The comparison can result in a pruned filter 13. The pruned filter 13 may be pruned because the sum of absolute kernel weights of the kernel for that filter is below a threshold, or because it has the lower sum compared to the other filters in the kernel matrix 10. Because the pruned filter 13 is removed, the corresponding feature map 23 is also removed from the subsequent layer of feature maps 2.

[0061] In the kernel matrix 20, as with the kernel matrix 10, the vertical axis relates to an input channel 22. Because the input channel 22 of this subsequent kernel matrix 20 corresponds to the output channel 12 of the previous kernel matrix 10, the removal of the filter 13 is reflected in kernel matrix 20 by the removal of a row that corresponds to the column of the pruned filter 13 in kernel matrix 10 and the feature map 23.

[0062] Accordingly, the pruning of a filter 13 results in the removal the corresponding kernels of kernel matrix 10, as well as the subsequent corresponding feature map 23 and the corresponding kernels in kernel matrix 20 to the pruned filter 13.

[0063] Referring now to FIG. 5, a system/method for pruning intermediate filters of a simple convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0064] According to an embodiment of the present invention, filter 33 pruned from a kernel matrix 30 is illustrated. Pruning includes a set of intermediate feature maps 4 and subsequent intermediate and/or output feature maps 5. The intermediate feature maps 4 have filters applied with kernel matrix 30 to generate subsequent feature maps 5.

[0065] In this example, feature map 44 has already been pruned from intermediate feature maps 4. Accordingly, as discussed above, the feature maps 4 being input into input channel 31 results in a row of kernels for filter 34 in kernel matrix 30 being removed corresponding to the removed feature map 44.

[0066] The output channel 32 is then pruned by determining the absolute kernel weight sum for each filter (column) in the kernel matrix 30. Because the filter 34 has been removed from the kernel matrix 30, the weights of the kernels in that filter are not considered for the calculation of absolute kernel weight sums. The result of the calculation of absolute kernel weight sums permits the comparison of each filter with a threshold, as discussed above. As a result, filter 33 may be pruned as being below the threshold.

[0067] Because filter 33 for the output channel 32 has been removed, the corresponding feature map 53 of subsequent feature maps 5 will also be removed. Thus, a feature map from each of feature maps 4 and 5 are removed according to the pruning process, thus reducing convolutional operations accordingly, and reducing the requirement resources of employing the CNN.

[0068] Referring now to FIG. 6, a system/method for pruning intermediate filters of a residual convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0069] According to an embodiment of the present invention, a more complicated CNN may be pruned, such as, e.g., a residual CNN (for example, RESNET). In such a CNN, The RESNET CNN will include a projection shortcut kernel matrix 60 and a residual block kernel matrix 40. Intermediate feature maps 6 may include an already pruned feature map 63. To prune subsequent filters, the feature maps 6 may be filtered with both the projection shortcut kernel matrix 60 and the residual block kernel matrix 40 independently, thus creating two separate sets of feature maps, projection shortcut feature maps 8b and residual block feature maps 7.

[0070] The filters of kernel matrix 40 may be pruned to remove pruned filters 43 according to systems and methods discussed above. As a result, the corresponding feature maps 73 are removed from feature maps 7. However, filters of a subsequent kernel matrix 50 are not pruned according to a threshold comparison of the absolute kernel weight sums of filters in the kernel matrix 50. Rather, pruned filters 53 are determined according to an analysis of the projection shortcut kernel matrix 60. Accordingly, the filters of the projection shortcut kernel matrix 60 under absolute kernel weight summation and comparison to a threshold according to

aspects of the invention described above. As a result, pruned filters 53 are determined to be pruned from the projection shortcut kernel matrix 60.

[0071] Therefore, corresponding projection shortcut feature maps 8b are generated in parallel to the residual block feature maps 7 and reflecting the removal of pruned filters 53. As a result of the removal of pruned filters 53, the corresponding projection shortcut feature maps 83b are removed from the projection shortcut feature maps 8b. The projection shortcut feature maps 8b are used to determine pruned filters 53 in the subsequent residual block kernel matrix 50. Accordingly, resulting subsequent residual block feature maps 8a are updated to reflect the removal of pruned subsequent residual feature maps 83a corresponding to the pruned filter 53.

[0072] Referring now to FIG. 7, a high-level system/method for surveillance of forest fires with image recognition using a pruned convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0073] In one embodiment, a system is contemplated for providing surveillance for a forest 740. According to aspects of the present invention, surveillance may include image recognition to facilitate the recognition and prediction of potentially dangerous or hazardous situations 742, such as, e.g., fires, floods, or any other natural or man-made condition in the forest 740.

[0074] Surveillance may be performed by one or more image capture devices 701. The image capture devices 701 is used to capture an image or sequence of images from a remote location of the forest 740 such that the image or sequence of images may be analyzed to recognize and predict a hazardous situation, such as, e.g., a wildfire 742. Accordingly, the image capture devices 701 may each include, e.g., a still or video camera mounted in a remote location, such as, e.g., a fixed tower or an autonomous or remotely operated drone. The image capture devices 701 can also include any other device incorporating a suitable sensor for capturing images (for example, e.g., a device including a charge-coupled device (CCD), a photodiode, a complementary metal oxide semiconductor (CMOS) sensor, infrared sensor, laser distance and ranging (LIDAR) sensor, among others). The remote image capture devices 701 can, therefore, include batteries for providing power to components.

[0075] The image capture devices 701 may each include a processing system 700 for performing the analysis, including image recognition. The processing system 700 may be an internal component to the image capture device 701, or may be external and in communication to the image capture device 701. Accordingly, the processing system 700 may include, e.g., a system-on-chip (SOC), a computer processing unit (CPU), a graphical processing unit (GPU), a computer system, a network, or any other processing system.

[0076] The processing system 700 may include a computer processing device 710 for processing the image or sequence of images provided by the image capture devices 701. Accordingly, the computer processing device 710 can include, e.g., a CPU, a GPU, or other processing device or combination of processing devices. According to aspects of an embodiment of the present invention, the processing system 700 is an embedded processing system including, e.g., a SOC.

[0077] Such systems have strict power and resource constraints because they do not benefit from grid power or large packaging then provide a large volume of power, storage or processing power. Therefore, to perform the image recognition, the computer processing device **710** includes a pruned CNN **712**. The pruned CNN **712** reduces the complexity of the CNN for image recognition by pruning particular portions of the CNN that have a relatively insignificant impact on the accuracy of the image recognition. Thus, pruning a CNN can reduce the resource requirements of the CNN by reducing complexity, while having minimal impact on the accuracy of the CNN. As a result, the pruned CNN **712** may be, e.g., stored in volatile memory, such as, e.g., dynamic random access memory (DRAM), processor cache, random access memory (RAM), or a storage device, among other possibilities. Accordingly, the pruned CNN **712** may be stored locally to an embedded or mobile processing system and provide accurate image recognition results.

[0078] The image recognition results from the pruned CNN **712** can be combined with action prediction to predict changes to situations. Thus, the processing system **700** analyzes the recognized images from the pruned CNN **712** to predict future movement and actions of features in the image or images. Thus, forest fires, for example, can be predicted and proactively addressed. The action prediction can be performed by the pruned CNN **712**, or, e.g., by additional features such as, e.g., recurrent neural networks (RNN) including long short-term memory, among other prediction solutions.

[0079] The results, including the image recognition results and any predictions, can be communicated from the computer processing system **710** to a transmitter **720**. The transmitter **720** may be included in the processing system **700**, or it may be separate from the processing system **700**. The transmitter **720** communicates the image recognition results to a remote location or device via a receiver **722** at that location or device. The communication may be wired or wireless, or by any suitable means to provide the image recognition results.

[0080] Upon receipt of the image recognition results, the receiver **722** communicates the results to a notification system **730**. The notification system **730** provides a user with a notification of situations predicted with the recognized images, such as, an alert of a hazardous situation such as, e.g. a wildfire **742** by, e.g., a display of the recognized images and predicted situations including, e.g., present or predicted labelled images, object lists, actions, or conditions. Accordingly, the notification system **730** may take the form of, e.g., a display, a speaker, an alarm, or combinations thereof. Thus, a user may be notified of the features that have been recognized and predicted from an image by the pruned CNN **712**, and act accordingly.

[0081] Moreover, the multiple remote image capture devices **701** can be deployed to perform image recognition and prediction in a single area. The results from each of the image capture devices **701** can then be cross referenced to determine a level of confidence of the results. Thus, accuracy of the image recognition and prediction can be improved using multiple remote image capture devices **701**.

[0082] Referring now to FIG. **8**, a high-level system/method for surveillance for vehicles with image recognition using a pruned convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0083] In one embodiment, a system is contemplated for providing surveillance for a vehicle **830**, such as, e.g., an autonomous vehicle, a semi-autonomous vehicle, an autonomous drone, a user operated vehicle, among others. According to aspects of the present invention, surveillance may include image recognition to facilitate the recognition and response to potentially dangerous or hazardous situations, such as, e.g., pedestrians **840**, obstacles, other vehicles, lane line, among others.

[0084] Surveillance may be performed by an image capture device **801**. The image capture device **801** is used to capture an image or sequence of images of the vehicle **830** such that the image or sequence of images may be analyzed to recognize or predict route obstacles. Accordingly, the image capture device **801** may include, e.g., a still or video camera, or anything device incorporating a suitable sensor for capturing images (for example, e.g., a device including a charge-coupled device (CCD), a photodiode, a complementary metal oxide semiconductor (CMOS) sensor, infrared sensor, laser distance and ranging (LIDAR) sensor, among others). Moreover, the image capture device **801** can be included with the vehicle **840**, or it can be a separate device that is mountable to the vehicle **830**. In any case, the image capture device **801** can include a power source, such as, e.g., a battery, or it can be powered by a battery included in the vehicle **830**.

[0085] The image capture device **801** may include a processing system **800** for performing the analysis, including image recognition. The processing system **800** may be an internal component to the image capture device **801**, or may be external and in communication to the image capture device **801**. Accordingly, the processing system **800** may include, e.g., a system-on-chip (SOC), a computer processing unit (CPU), a graphical processing unit (GPU), a computer system, a network, or any other processing system.

[0086] The processing system **800** may include a computer processing device **810** for processing the image or sequence of images provided by the image capture device **801**. Accordingly, the computer processing device **810** can include, e.g., a CPU, a GPU, or other processing device or combination of processing devices. According to aspects of an embodiment of the present invention, the processing system **800** is an embedded processing system including, e.g., a SOC, or a mobile device such as, e.g., a smartphone.

[0087] Such systems have strict power and resource constraints because they do not benefit from grid power or large packaging then provide a large volume of power, storage or processing power. Therefore, to perform the image recognition, the computer processing device **810** includes a pruned CNN **812**. The pruned CNN **812** reduces the complexity of the CNN for image recognition by pruning particular portions of the CNN that have a relatively insignificant impact on the accuracy of the image recognition. Thus, pruning a CNN can reduce the resource requirements of the CNN by reducing complexity, while having minimal impact on the accuracy of the CNN. As a result, the pruned CNN **812** may be, e.g., stored in volatile memory, such as, e.g., dynamic random access memory (DRAM), processor cache, random access memory (RAM), or a storage device, among other possibilities. Accordingly, the pruned CNN **812** may be stored locally to an embedded or mobile processing system and provide accurate image recognition results.

[0088] The image recognition results from the pruned CNN **812** can be combined with action prediction to predict

changes to situations. Thus, the processing system **800** analyzes the recognized image from the pruned CNN **812** to predict future movement and actions of features in the image or images. Thus, obstacles in the road, such as, e.g., pedestrians, downed trees, animals, other vehicles, etc. can be predicted and proactively addressed. The action prediction can be performed by the pruned CNN **812**, or, e.g., by additional features such as, e.g., recurrent neural networks (RNN) including long short-term memory, among other prediction solutions.

[0089] The results, including the image recognition results and any predictions, can be communicated from the computer processing system **810** to a transmitter **820**. The transmitter **820** may be included in the processing system **800**, or it may be separate from the processing system **800**. The transmitter **820** communicates the image recognition results to a receiver **804**. The communication may be wired or wireless, or by any suitable means to provide the image recognition results. The receiver **804** may be external to the vehicle **830**, or it may be a part of the vehicle **830**. Thus, the computer processing system **810** can send the results to, e.g., a remote operator, a local operator, or to the vehicle **830** itself.

[0090] Upon receipt of the image recognition results, the receiver **804** can communicate the results to a notification system **806**. The notification system **806** provides a user with a notification of the recognized images, such as a pedestrian **840** by, e.g., a display of the recognized images and predicted obstacles including, e.g., present or predicted labelled images, object lists, actions, or conditions. Accordingly, the notification system **806** may take the form of, e.g., a display, a speaker, an alarm, or combinations thereof. Thus, a user may be notified of the features that have been recognized and obstacles predicted by the pruned CNN **812**, and act accordingly. The notification system **806** may be at a remote location for a remote device operator, or it may be internal to the vehicle, such as in the case of notifying an operator of the vehicle. Additionally, the processing system **800** of the vehicle **830** can be programmed to automatically take action to avoid a recognized or predicted obstacle by, e.g., turning, stopping, or taking any other suitable evasive action.

[0091] Referring now to FIG. 9, a flow diagram illustrating a system/method for image recognition using a pruned convolution neural network (CNN) is illustratively depicted in accordance with an embodiment of the present principles.

[0092] At block **901**, feature maps and corresponding convolution filters are extracted from a convolutional neural network (CNN) for surveillance image recognition.

[0093] The CNN is a fully trained CNN for image recognition. Through the training and testing process of the CNN, the convolution layers can be extracted. Each convolutional layer includes a set of feature maps with a corresponding kernel matrix. The feature maps have corresponding filters that influence the generation of the feature maps according to the kernel weights for each filter. As discussed above, a filter may be a 3-dimensional (3D) filter having a kernel at each input channel of a kernel matrix. Thus, a set of kernels, one from each input channel, forms a filter for an output channel of the kernel matrix. The filter is used to generate an output feature map. The kernel matrix includes a plurality of filters, each corresponding to a particular feature map.

[0094] At block **902**, absolute magnitude kernel weights are summed for each convolutional filter.

[0095] As discussed above, each filter includes a kernel from each input channel of the kernel matrix. The magnitude of a filter can be represented by calculating the absolute kernel weight of each kernel, and then summing those weights for each filter. The magnitude of a filter is indicative of the frequency with which a filter is activated for image recognition.

[0096] At block **903**, low magnitude convolutional filters and corresponding feature maps are removed according to the absolute magnitude kernel weight sums.

[0097] Because convolutional operations are very computationally costly, the removal of convolutional filters, which reduces convolutional operations, can reduce the computation cost of the CNN. Therefore, upon determining the magnitude of each filter in a layer, filters with low magnitudes can be removed.

[0098] The filters to be removed can be determined by a comparison with a threshold value. For example, filters having a magnitude below a certain threshold can be removed. As another example, a certain number of the smallest filters can be removed. Other possible thresholds are contemplated.

[0099] Alternatively, or in addition, sensitivity to pruning may be a determining factor for whether a filter is pruned. Sensitivity may be assessed by, e.g., removing a filter and testing the effect of the removal on accuracy to determine if it exceeds a threshold value at which a filter is deemed too sensitive to pruning, and thus is replaced into the kernel matrix, or by comparing the weights of each kernel in the filter to a threshold magnitude, and if there are too few kernel weight weights with a magnitude below the threshold then the filter is deemed sensitive to pruning and should not be removed.

[0100] Because a filter is removed, the corresponding output feature map from that filter should be removed as well. Thus, the feature map set at a given layer is reduced in size. This reduction results in a corresponding reduction in resource requirements for implementing the CNN.

[0101] At block **904**, kernels of a subsequent convolutional layer corresponding to the removed feature maps are removed.

[0102] The removal of a particular filter and feature map affects subsequent kernel matrices. In particular, a filter, including a set of kernels for an output channel in the kernel matrix of a given layer, that is removed from the kernel matrix results in the corresponding kernels being absent from the input channel of a kernel matrix in a subsequent layer. As a result, the subsequent kernel matrix should be updated with the kernels of the removed filter pruned so that the pruning of the subsequent kernel matrix is not influenced by the removed kernels.

[0103] At block **905**, the CNN is retrained with updated filters.

[0104] Because filters have been removed, the CNN can be retrained to compensate for the absent filters. Because the removed filters have low impact on the image recognition task as indicated by the low magnitudes, the CNN can be retrained without the filters to regain the original accuracy of the trained CNN prior to pruning. Thus, a CNN may be pruned to reduce computation resources by removing convolutional operations associated with filter operations while not negatively impacting the accuracy of the CNN.

[0105] 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 method for pruning a convolutional neural network (CNN) for surveillance with image recognition, the method comprising:

extracting at least one convolutional layer from a trained CNN, each convolutional layer including a kernel matrix having at least one filter formed in a corresponding output channel of the kernel matrix, and feature map set having a feature map corresponding to each of the at least one filter;

determining an absolute kernel weight for each kernel in the kernel matrix;

summing the absolute kernel weights of each kernel in each of the at least one filter to determine a magnitude of each filter;

comparing the magnitude of each filter with a threshold and removing one or more filters that are below the threshold;

removing a feature map corresponding to each of the removed filters to prune the CNN of filters; and

retraining the CNN upon pruning the removed filters to generate a pruned CNN having fewer convolutional layers to efficiently recognize and predict conditions in an environment being surveilled.

2. The method as recited in claim 1, further comprising removing the kernels of the removed filter from subsequent kernel matrices.

3. The method as recited in claim 2, further comprising, upon removing the kernels of the removed filter from subsequent kernel matrices, pruning the subsequent kernel matrices.

4. The method as recited in claim 1, further comprising iteratively retraining the CNN upon pruning each convolutional layer.

5. The method as recited in claim 1, further comprising retraining the CNN upon pruning every convolutional layer.

6. The method as recited in claim 1, wherein the threshold is a value corresponding to a minimum absolute kernel weight sum.

7. The method as recited in claim 1, wherein the threshold is a value corresponding to a number of filters having the smallest absolute kernel weight sums to be removed.

8. A non-transitory computer readable storage medium comprising a computer readable program for surveillance with image recognition using a pruned convolutional neural network (CNN), wherein the computer readable program when executed on a computer causes the computer to perform the steps of:

extracting at least one convolutional layer from a trained CNN, each convolutional layer including a kernel matrix having at least one filter formed in a correspond-

ing output channel of the kernel matrix, and feature map set having a feature map corresponding to each of the at least one filter;

determining an absolute kernel weight for each kernel in the kernel matrix;

summing the absolute kernel weights of each kernel in each of the at least one filter to determine a magnitude of each filter;

comparing the magnitude of each filter with a threshold and removing one or more filters that are below the threshold;

removing a feature map corresponding to each of the removed filters to prune the CNN of filters; and

retraining the CNN upon pruning the removed filters to generate a pruned CNN having fewer convolutional layers to efficiently recognize and predict conditions in an environment being surveilled.

9. The computer readable program as recited in claim 8, further comprising removing the kernels of the removed filter from subsequent kernel matrices.

10. The computer readable program as recited in claim 9, further comprising, upon removing the kernels of the removed filter from subsequent kernel matrices, pruning the subsequent kernel matrices.

11. The computer readable program as recited in claim 8, further comprising iteratively retraining the CNN upon pruning each convolutional layer.

12. The computer readable program as recited in claim 8, further comprising retraining the CNN upon pruning every convolutional layer.

13. The computer readable program as recited in claim 8, wherein the threshold is a value corresponding to a minimum absolute kernel weight sum.

14. The computer readable program as recited in claim 8, wherein the threshold is a value corresponding to a number of filters having the smallest absolute kernel weight sums to be removed.

15. An image recognition system for surveillance, the system comprising:

an image capture device for capturing images of an environment to be surveilled;

an image recognition system in an embedded computing device included in the image capture device configured to perform image recognition with a pruned CNN, the image recognition system including:

an absolute kernel weight summer configured to determining an absolute kernel weight for each kernel in the kernel matrix and sum the absolute kernel weights of each kernel in a filter corresponding to each of at least one output channel of a kernel matrix to determine a magnitude of each filter, each filter corresponding to a feature map;

a threshold comparison unit configured to comparing the magnitude of each filter with a threshold and removing one or more filters that are below the threshold;

a layer updater configured to removing a feature map corresponding to each of the removed filters to prune the CNN of filters and generate the pruned CNN;

a long short-term memory network (LSTM) for predicting feature actions;

an action network for generating class probabilities of feature actions; and

a notification device for notifying a user of the class probabilities.

16. The system as recited in claim **15**, wherein the layer update is further configured to remove the kernels of the removed filter from subsequent kernel matrices.

17. The system as recited in claim **15**, wherein the image recognition system is further configured to iteratively retrain the CNN upon pruning each convolutional layer.

18. The system as recited in claim **15**, wherein the image recognition system is further configured to retrain the CNN upon pruning every convolutional layer.

19. The system as recited in claim **15**, wherein the threshold is a value corresponding to a minimum absolute kernel weight sum.

20. The system as recited in claim **15**, wherein the threshold is a value corresponding to a number of filters having the smallest absolute kernel weight sums to be removed.

* * * * *