(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 April 2024 (18.04.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/080929 A1

(51) International Patent Classification:
G06V 10/88 (2022.01) *G06Q 50/08* (2012.01)
G06N 3/0464 (2023.01)

(21) International Application Number:
PCT/SG2023/050685

(22) International Filing Date:
11 October 2023 (11.10.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10202251368T 13 October 2022 (13.10.2022) SG

(71) Applicant: **NANYANG TECHNOLOGICAL UNIVERSITY** [SG/SG]; 50 Nanyang Avenue, Singapore 639798 (SG).

(72) Inventors: **XIE, Lihua**; c/o Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798 (SG). **YANG, Jianfei**; c/o Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798 (SG). **QIAN, Hanjie**; c/o Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798 (SG).

(74) Agent: **DAVIES COLLISON CAVE ASIA PTE LTD**; Attn: Mr. KOH, Wen Chin (Aaron), 10 Collyer Quay #07-01, Ocean Financial Centre, Singapore 049315 (SG).

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

(54) Title: METHOD AND SYSTEM FOR TRAINING A NEURAL NETWORK MODEL TO PERFORM IMAGE CLASSIFICATION AND IMAGE LOCALIZATION AND/OR SEGMENTATION

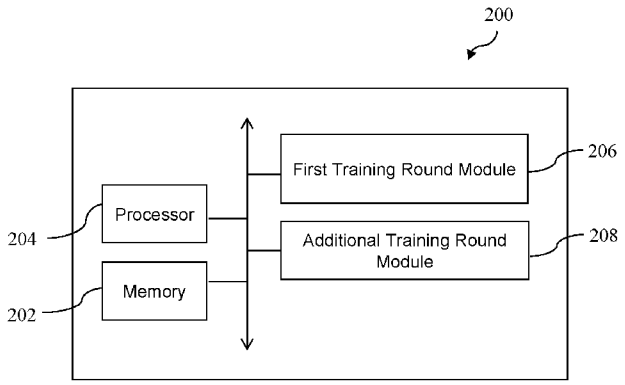


FIG. 2

(57) Abstract: A method of training a neural network model is provided. The neural network model includes a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or segmentation. The method includes: training, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters; applying, at the first training round, the first backbone parameters obtained corresponding to the first training round to the second neural network portion; training, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters; and performing, after the first training round, a plurality of additional training rounds. Each additional training round includes: updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained; training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters; updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained; and training, at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters.

WO 2024/080929 A1

[Continued on next page]

RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

Published:

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

METHOD AND SYSTEM FOR TRAINING A NEURAL NETWORK MODEL TO PERFORM IMAGE CLASSIFICATION AND IMAGE LOCALIZATION AND/OR SEGMENTATION

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of priority of Singapore Patent Application No. 10202251368T filed on 13 October 2022, the content of which being hereby incorporated by reference in its entirety for all purposes.

TECHNICAL FIELD

[0002] The present invention generally relates to a method of training a neural network model to perform image classification and image localization and/or segmentation, and a system thereof, such as with respect to structural images for building defect detection.

BACKGROUND

[0003] There is a wide range of practical applications in which image classification and image localization and/or segmentation may be desired or required. An example practical application is in vision-based structural health monitoring for building defect detection, including building defect classification and building defect localization and/or segmentation.

[0004] Conventional approaches are to train separate neural network models for image classification tasks and the image localization and/or segmentation tasks. That is, a neural network model for performing image classification is trained based on a training dataset with labelled images for image classification and another neural network model for performing image localization and/or segmentation is separately trained based on another training dataset with labelled images for image localization and/or segmentation.

[0005] There are a number of drawbacks associated with such conventional approaches. Firstly, training separate neural network models for different types of tasks is not only inefficient (e.g., the need to train these neural network models separately) but also inconvenient/cumbersome (e.g., two separate neural network models need to be deployed to perform the different types of tasks). In addition, when such separate neural network models are deployed to perform image classification and image localization and/or segmentation for a practical application, the overall performance can be unsatisfactory since relationships/correlations between different types of tasks associated with the practical

application are not captured. Still further, when such separate neural network models are employed in practical applications, there can be a significant drop in performances when compared with those obtained based on training data.

[0006] A need therefore exists to provide a method of training a neural network model to perform image classification and image localization and/or segmentation, as well as a system thereof, that seeks to overcome, or at least ameliorate, one or more deficiencies in conventional methods or approaches, and more particularly, with improved/enhanced training efficiency and effectiveness, as well as producing a trained neural network model with improved/enhanced usability and/or performances in image classification and image localization and/or segmentation in practical applications, such as with respect to structural images for building defect detection. It is against this background that the present invention has been developed.

SUMMARY

[0007] According to a first aspect of the present invention, there is provided a method of training a neural network model using at least one processor, the neural network model comprising a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or image segmentation, the method comprising:

training, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round;

applying, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion;

training, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round; and

performing, after the first training round, a plurality of additional training rounds, each additional training round comprising:

updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round;

training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion corresponding to the additional training round;

updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round; and

training, at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round.

[0008] According to a second aspect of the present invention, there is provided a system for training a neural network model, the neural network model comprising a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or image segmentation, the system comprising:

at least one memory; and

at least one processor communicatively coupled to the at least one memory and configured to:

train, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round;

apply, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion;

train, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round; and

perform, after the first training round, a plurality of additional training rounds, each additional training round comprising:

updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round;

training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion corresponding to the additional training round;

updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round; and

training, at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round.

[0009] According to a third aspect of the present invention, there is provided a computer program product, embodied in one or more non-transitory computer-readable storage mediums, comprising instructions executable by at least one processor to perform the method of training a neural network model according to the above-mentioned first aspect of the present invention.

[0010] According to a fourth aspect of the present invention, there is provided a method of using the neural network model trained according to the above-mentioned first aspect of the present invention, the method comprising:

receiving, by the neural network model, an input image;

performing, by the neural network model, image classification of the input image; and

performing, by the neural network model, image localization and/or image segmentation of the input image,

wherein the above-mentioned performing image classification of the input image and the above-mentioned performing image localization and/or image segmentation of the input image are performed simultaneously.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] Embodiments of the present invention will be better understood and readily apparent to one of ordinary skill in the art from the following written description, by way of example only, and in conjunction with the drawings, in which:

FIG. 1 depicts a schematic flow diagram of a method of training a neural network model, according to various embodiments of the present invention;

FIG. 2 depicts a schematic block diagram of a system for training a neural network model, according to various embodiments of the present invention;

FIG. 3 depicts a schematic block diagram of an exemplary computer system which may be used to realize or implement the system for training a neural network model, according to various embodiments of the present invention;

FIG. 4 depicts a hierarchical structure (ϕ-NeXt) for a set of classification tasks, according to various example embodiments of the present invention;

FIG. 5 shows a number of example multi-attribute structural images, along with the attributes associated with each structural image, according to various example embodiments of the present invention;

FIG. 6 depicts an example interface for a labeling tool for labeling images, according to various example embodiments of the present invention;

FIG. 7 depicts a bar graph showing the statistics of training and test label distributions for classification tasks 1 to 8, according to various example embodiments of the present invention;

FIG. 8A shows a table (Table 1) including performance results obtained for a number of models under the baseline, according to various example embodiments of the present invention;

FIG. 8B shows a table (Table 2) including performance results obtained for a number of models under the transfer learning approach, according to various example embodiments of the present invention;

FIG. 9 shows four example simplified hierarchical transfer learning paths, according to various example embodiments of the present invention;

FIG. 10 shows a table (Table 3) including performance results obtained for a model under the hierarchical transfer learning approach, according to various example embodiments of the present invention;

FIG. 11 depicts a schematic drawing of a multi-task learning (MTL) framework or model, according to various example embodiments of the present invention;

FIG. 12 shows a table (Table 4) including performance results obtained for a model under the multi-task learning approach, according to various example embodiments of the present invention;

FIG. 13 depicts a schematic drawing of the multi-task heterogeneous learning framework or model, according to various example embodiments of the present invention;

FIG. 14 depicts a schematic drawing of an example architecture of the neural network model shown in FIG. 13, according to various example embodiments of the present invention;

FIG. 15 shows a table (Table 5) including localization and segmentation results obtained for the heterogeneous learning model using a backbone sharing approach, according to various example embodiments of the present invention;

FIG. 16 shows the performances of (1) a first model trained based on the ϕ-Net dataset and evaluated on the ϕ-Net dataset, (2) a second model trained based on the Residential dataset and evaluated on the Residential dataset, and (3) the first model evaluated on the Residential dataset, with respect to an example task, according to various example embodiments of the present invention;

FIG. 17 depicts a schematic drawing of the heterogeneous learning framework or model that is further subjected to environment adaptive learning, according to various example embodiments of the present invention;

FIG. 18 shows a plot of the model performance against the value of the alignment coefficient K, according to various example embodiments of the present invention;

FIGs. 19A and 19B show a one-dimensional signal and a Fourier transform of the one-dimensional signal, respectively, according to various example embodiments of the present invention;

FIGs. 20A and 20B show an example image before and after Fourier Transform, according to various example embodiments of the present invention; and

FIG. 21A shows a table (Table 6) comparing the classification performances (accuracy %) of the original model and the environment adaptive model on the residential dataset;

FIG. 21B shows a table (Table 7) comparing the localization performances (accuracy %) of the original model and the environment adaptive model on the residential dataset; and

FIG. 21C shows a table (Table 8) comparing the segmentation performances (accuracy %) of the original model and the environment adaptive model on the residential dataset.

DETAILED DESCRIPTION

[0012] Various embodiments of the present invention provide a method and a system for training a neural network model to perform image classification and image localization and/or segmentation, such as with respect to structural images for building defect detection.

[0013] As explained in the background, conventional approaches are to train separate neural network models for image classification tasks and image localization and/or segmentation tasks. However, various embodiments note that there are a number of drawbacks associated with such conventional approaches, including being inefficient (e.g., the need to train these neural network models separately), being inconvenient/cumbersome (e.g., multiple separate neural

network models need to be deployed to perform the different types of tasks) and producing unsatisfactory performances (e.g., relationships/correlations between different types of tasks associated with a practical application not being captured and performance degradations when deployed in practical applications compared with those obtained based on training data). In this regard, various embodiments of the present invention provide a method of training a neural network model to perform image classification and image localization and/or segmentation, as well as a system thereof, that seeks to overcome, or at least ameliorate, one or more deficiencies in conventional methods or approaches, and more particularly, with improved/enhanced training efficiency and effectiveness, as well as producing a trained neural network model with improved/enhanced usability and/or performances in image classification and image localization and/or segmentation in practical applications, such as with respect to structural images for building defect detection.

[0014] FIG. 1 depicts a schematic flow diagram of a method 100 of training a neural network model using at least one processor, the neural network model comprising a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or segmentation. The method 100 comprises: training (at 106), at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round; applying (at 108), at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion; and training (at 110), at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round. The method 100 further comprises performing (at 112), after the first training round, a plurality of additional training rounds. Each additional training round comprises: updating (at 112a), at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round (e.g., based on the second backbone parameters obtained corresponding to the first training round if there is no previous additional training round with respect to the current additional training round or otherwise based on the second backbone parameters obtained corresponding to the immediately previous additional training round); training (at 112b), at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion

corresponding to the additional training round; updating (at 112c), at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round; and training (at 112d), at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round.

[0015] Accordingly, the method 100 of training a neural network model according to various embodiments of the present invention advantageously has improved/enhanced training efficiency and effectiveness. In this regard, the first neural network portion configured to perform image classification and the second neural network portion configured to perform image localization and/or segmentation are advantageously trained as one neural network model (as a whole), as well as deployed as one neural network model to perform image classification and image localization and/or segmentation. Furthermore, the first and second neural network portions are trained in a manner which facilitates or enables the sharing of their backbone parameters (e.g., weight parameters) therebetween, which has been found to enhance performances of the neural network model in image classification and image localization and/or segmentation in practical applications. These advantages or technical effects, and/or other advantages or technical effects, will become more apparent to a person skilled in the art as the method 100 of training a neural network model, as well as the corresponding system for training a neural network model, is described in more detail according to various embodiments and example embodiments of the present invention.

[0016] In various embodiments, the above-mentioned updating (at 112a), at the additional training round, the first backbone parameters of the first neural network portion is further based on a first backbone sharing parameter. In various embodiments, similarly, the above-mentioned updating (at 112c), at the additional training round, the second backbone parameters of the second neural network portion is further based on a second backbone sharing parameter.

[0017] In various embodiments, the first backbone sharing parameter is configured to control an amount of the second backbone parameters obtained for updating the first backbone parameters of the first neural network portion. In various embodiments, similarly, the second backbone sharing parameter is configured to control an amount of the first backbone parameters obtained for updating the second backbone parameters of the second neural network portion.

[0018] In various embodiments, the above-mentioned performing (at 112) the plurality of additional training rounds comprises performing a number of additional training rounds until each of the first and second neural network portions converges.

[0019] In various embodiments, the first and second neural network portions each comprises a Swin Transformer as a backbone thereof.

[0020] In various embodiments, the first backbone parameters comprise first weight parameters. In various embodiments, the second backbone parameters comprise second weight parameters.

[0021] In various embodiments, the first neural network portion comprises a plurality of task classifiers configured to perform a plurality of classification tasks, respectively. In this regard, the training dataset comprises labelled images comprising multi-attribute labelled images, and the first neural network portion is trained for the plurality of classification tasks simultaneously (e.g., as a whole instead of separately).

[0022] In various embodiments, the method 100 further comprises performing fine-tuning of the first and second neural network portions comprising: performing, for each labelled image of a subset of labelled images of the training dataset, a low-frequency spectral alignment of the labelled image with respect to an image of a subset of images of a second dataset to obtain a low-frequency aligned labelled image, thereby obtaining a low-frequency aligned training data subset.

[0023] In various embodiments, the above-mentioned performing, for each labelled image of the subset of labelled images of the training dataset, the low-frequency spectral alignment of the labelled image with respect to the image of the subset of images of the second dataset comprises: obtaining a Fourier transform of the labelled image and a Fourier transform of the image; obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the labelled image and obtaining a magnitude spectrum of the Fourier transform of the image; and performing the low-frequency spectral alignment of the labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image, the magnitude spectrum of Fourier transform of the image and a first alignment parameter.

[0024] In various embodiments, the above-mentioned performing the low-frequency spectral alignment of the labelled image comprises: modifying the magnitude spectrum of the Fourier transform of the labelled image based on the magnitude spectrum of the Fourier transform of the image and the first alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the labelled image; and performing an inverse Fourier

transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image to obtain the low-frequency aligned labelled image.

[0025] In various embodiments, the first alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the image obtained for modifying the magnitude spectrum of the Fourier transform of the labelled image. In various embodiments, the first alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

[0026] In various embodiments, the above-mentioned performing fine-tuning of the first and second neural network portions further comprises: performing, for each image of a plurality of images of the subset of images of the second dataset, self-supervised labelling of the image to obtain a pseudo-labelled image, thereby obtaining a subset of pseudo-labelled images; and performing, for each pseudo-labelled image of the subset of pseudo-labelled images, a low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the training dataset to obtain a low-frequency aligned pseudo-labelled image, thereby obtaining a low-frequency aligned second data subset.

[0027] In various embodiments, the above-mentioned performing, for each pseudo-labelled image of the subset of pseudo-labelled images, the low-frequency spectral alignment of the pseudo-labelled image with respect to the labelled image of the subset of labelled images of the training dataset comprises: obtaining a Fourier transform of the pseudo-labelled image and a Fourier transform of the labelled image; obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the pseudo-labelled image and obtaining a magnitude spectrum of the Fourier transform of the labelled image; and performing the low-frequency spectral alignment of the pseudo-labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image, the magnitude spectrum of Fourier transform of the labelled image and a second alignment parameter.

[0028] In various embodiments, the above-mentioned performing the low-frequency spectral alignment of the pseudo-labelled image comprises: modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image based on the magnitude spectrum of the Fourier transform of the labelled image and the second alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the pseudo-labelled image; and performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image to obtain the low-frequency aligned pseudo-labelled image.

[0029] In various embodiments, the second alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the labelled image obtained for modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image. In various embodiments, the second alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

[0030] In various embodiments, the above-mentioned performing fine-tuning of the first and second neural network portions comprises performing a plurality of fine-tuning rounds. Each of the plurality of fine-tuning rounds comprises: shuffling the labelled images in the training dataset and the images in the second dataset; extracting a plurality of subsets of images from the second dataset, and for each subset of images, extracting a subset of labelled images from the training dataset to form a subset pair of the subset of images and the subset of labelled images, thereby forming a plurality of subset pairs; and for each subset pair of the plurality of subset pairs: performing, for each labelled image of the subset of labelled images of the subset pair, the above-mentioned low-frequency spectral alignment of the labelled image with respect to an image of the subset of images of the subset pair to obtain the low-frequency aligned labelled image, thereby obtaining the low-frequency aligned training data subset; performing, for each image of a plurality of images of the subset of images of the subset pair, the above-mentioned self-supervised labelling of the image to obtain the pseudo-labelled image, thereby obtaining the subset of pseudo-labelled images; performing, for each pseudo-labelled image of the subset of pseudo-labelled images, the above-mentioned low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the subset pair to obtain the low-frequency aligned pseudo-labelled image, thereby obtaining the low-frequency aligned second data subset; and training the first and second neural network portions based on the low-frequency aligned training data subset and the low-frequency aligned second data subset.

[0031] Accordingly, in various embodiments, the first and second neural network portions are further fine-tuned based on low-frequency aligned training data subsets and low-frequency aligned second data subsets (e.g., low-frequency aligned testing data subsets), which has been found to improve/enhance the usability of the neural network model in image classification and image localization and/or segmentation when deployed in practical applications.

[0032] In various embodiments, the frequency point is determined to correspond to a low-frequency point if the frequency point corresponds to a frequency component in a lowest 1% to

a lowest 10% of frequency components of an image being subjected to the low-frequency spectral alignment.

[0033] In various embodiments, the training dataset and the second dataset are obtained from different sources (e.g., from different physical environment).

[0034] In various embodiments, the second neural network portion is configured to perform building defect localization and/or building defect segmentation, and the training dataset comprises labelled structural images for training the first neural network portion to perform image classification and for training the second neural network portion to perform building defect localization and/or building defect segmentation.

[0035] FIG. 2 depicts a schematic block diagram of a system 200 for training a neural network model according to various embodiments of the present invention, corresponding to the above-mentioned method 100 of training a neural network model as described hereinbefore according with reference to FIG. 1 according to various embodiments of the present invention. The system 200 comprises: at least one memory 202; and at least one processor 204 communicatively coupled to the at least one memory 202 and configured to perform the method 100 of training a neural network model as described hereinbefore according to various embodiments of the present invention. Accordingly, the at least one processor 204 is configured to: train, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round; apply, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion; train, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round; and perform, after the first training round, a plurality of additional training rounds. Each additional training round comprising: updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round; training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion corresponding to the additional training round; updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round; and training, at the additional training round, the second neural network portion based

on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round.

[0036] It will be appreciated by a person skilled in the art that the at least one processor 204 may be configured to perform various functions or operations through set(s) of instructions (e.g., software modules) executable by the at least one processor 204 to perform various functions or operations. Accordingly, as shown in FIG. 2, the system 200 may comprise a first training round module (or a first training round circuit) 206 configured to: train, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round; apply, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion; and train, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round. The system 200 may further comprise an additional training round module (or an additional training round circuit) 208 configured to: perform, after the first training round, the above-mentioned plurality of additional training rounds.

[0037] It will be appreciated by a person skilled in the art that the above-mentioned modules are not necessarily separate modules, and two or more modules may be realized by or implemented as one functional module (e.g., a circuit or a software program) as desired or as appropriate without deviating from the scope of the present invention. For example, the first training round module 206 and the additional training round module 208 may be realized (e.g., compiled together) as one executable software program (e.g., software application or simply referred to as an “app”), which for example may be stored in the at least one memory 202 and executable by the at least one processor 204 to perform the corresponding functions or operations as described herein according to various embodiments.

[0038] In various embodiments, the system 200 for training a neural network model corresponds to the method 100 of training a neural network model as described hereinbefore with reference to FIG. 1, therefore, various operations, functions or steps configured to be performed by the least one processor 204 may correspond to various operations, functions or steps of the method 100 described hereinbefore according to various embodiments, and thus need not be repeated with respect to the system 200 for clarity and conciseness. In other words, various embodiments described herein in context of methods (e.g., the method 100 of training a neural network model) are analogously valid for the corresponding systems or devices (e.g.,

the system 200 for training a neural network model), and vice versa. For example, in various embodiments, the at least one memory 202 may have stored therein the first training round module 206 and the additional training round module 208, which respectively correspond to various operations, functions or steps of the method 200 of training a neural network model as described hereinbefore according to various embodiments, which are executable by the at least one processor 204 to perform the corresponding operations, functions or steps as described herein.

[0039] A computing system, a controller, a microcontroller or any other system providing a processing capability may be provided according to various embodiments in the present invention. Such a system may be taken to include one or more processors and one or more computer-readable storage mediums. For example, the system 200 described hereinbefore may include at least one processor (or controller) 204 and at least one computer-readable storage medium (or memory) 202 which are for example used in various processing carried out therein as described herein. A memory or computer-readable storage medium used in various embodiments may be a volatile memory, for example a DRAM (Dynamic Random Access Memory) or a non-volatile memory, for example a PROM (Programmable Read Only Memory), an EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM), or a flash memory, e.g., a floating gate memory, a charge trapping memory, an MRAM (Magnetoresistive Random Access Memory) or a PCRAM (Phase Change Random Access Memory).

[0040] In various embodiments, a "circuit" may be understood as any kind of a logic implementing entity, which may be special purpose circuitry or a processor executing software stored in a memory, firmware, or any combination thereof. Thus, in an embodiment, a "circuit" may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g., a microprocessor (e.g., a Complex Instruction Set Computer (CISC) processor or a Reduced Instruction Set Computer (RISC) processor). A "circuit" may also be a processor executing software, e.g., any kind of computer program, e.g., a computer program using a virtual machine code, e.g., Java. Any other kind of implementation of various functions or operations may also be understood as a "circuit" in accordance with various other embodiments. Similarly, a "module" may be a portion of a system according to various embodiments in the present invention and may encompass a "circuit" as above, or may be understood to be any kind of a logic-implementing entity therefrom.

[0041] Some portions of the present disclosure are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a

computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[0042] The present specification also discloses a system (e.g., which may also be embodied as one or more devices or apparatuses), such as the system 200, for performing various operations, functions or steps of various methods described herein. Such a system may be specially constructed for the required purposes or may comprise a general purpose computer system selectively activated or reconfigured by a computer program stored in the computer system. In general, various algorithms that may be presented herein are not limited to being implemented or executed by any particular computer system. Alternatively, the construction of more specialized computer system to perform various operations, functions or steps of various methods described herein may be provided as desired or as appropriate without going beyond the scope of the present invention.

[0043] In addition, the present specification also at least implicitly discloses computer program(s) or software/functional module(s), in that it would be apparent to a person skilled in the art that various operations, functions or steps of various methods described herein may be put into effect by computer code. The computer program(s) is not intended to be limited to any particular programming language and implementation thereof, and it will be appreciated by a person skilled in the art that a variety of programming languages and coding thereof may be used to implement the computer program(s). Moreover, the computer program(s) is not intended to be limited to any particular control flow as there are a variety of programming languages which can use different control flows. It will be appreciated by a person skilled in the art that a computer program may be stored on any computer-readable storage medium (non-transitory computer-readable storage medium), such as but not limited to, a magnetic disk, an optical disk or a memory chip. For example, a computer program stored on a computer-readable storage medium may be loaded and executed on a computer system to implement various operations, functions or steps of various methods described herein according to various embodiments of the present invention.

[0044] Accordingly, in various embodiments, there is provided a computer program product, embodied in one or more computer-readable storage mediums (non-transitory computer-readable storage medium), comprising instructions (e.g., the first training round module 206 and/or the additional training round module 208) executable by one or more computer processors to perform a method 100 of training a neural network model as described hereinbefore with reference to FIG. 1 according to various embodiments of the present invention. Accordingly, various computer programs or software modules described herein may be stored in a computer program product receivable by a system therein, such as the system 200 as shown in FIG. 2, for execution by at least one processor 204 of the system 200 to perform various operations, functions or steps of various methods described herein according to various embodiments of the present invention.

[0045] It will be appreciated by a person skilled in the art that various modules described herein (e.g., the first training round module 206 and/or the additional training round module 208) may be software module(s) realized by computer program(s) or set(s) of instructions executable by a computer processor to perform various functions or operations. Various modules described herein (e.g., the first training round module 206 and/or the additional training round module 208) may also be implemented as hardware module(s) being functional hardware unit(s) designed to perform various functions or operations. More particularly, in the hardware sense, a module is a functional hardware unit designed for use with other components or modules. For example, a module may be implemented using discrete electronic components, or it can form a portion of an entire electronic circuit such as an Application Specific Integrated Circuit (ASIC). Numerous other possibilities exist. It will also be appreciated by a person skilled in the art that a combination of hardware and software modules may be implemented. Furthermore, various operations, functions or steps of various methods described herein may be performed in parallel rather than sequentially as desired or as appropriate (e.g., as long as it does not render the method(s) inoperable or unsatisfactory for its intended purpose).

[0046] In various embodiments, the system 200 for training a neural network model may be realized by any computer system (e.g., desktop or portable computer system) including at least one processor and at least one memory, such as an example computer system 300 as schematically shown in FIG. 3 as an example only and without limitation. Various methods/steps or functional modules may be implemented as software, such as a computer program being executed within the computer system 300, and instructing the computer system

300 (in particular, one or more processors therein) to conduct various functions or operations as described herein according to various embodiments. The computer system 300 may comprise a system unit 302, one or more input devices 304 such as a keyboard, a touchscreen and/or a mouse, and a plurality of output devices such as a display 308. The system unit 32 may be connected to a computer network 312 via a suitable transceiver device 314, to enable access to e.g., the Internet or other network systems such as Local Area Network (LAN) or Wide Area Network (WAN). The system unit 302 may include a processor 318 for executing various instructions, a Random Access Memory (RAM) 320 and a Read Only Memory (ROM) 322. The system unit 302 may further include a number of Input/Output (I/O) interfaces, for example I/O interface 324 to the display device 308 and I/O interface 326 to the one or more input devices 304. The components of the system unit 302 typically communicate via an interconnected bus 328 and in a manner known to a person skilled in the art.

[0047] It will be appreciated by a person skilled in the art that the terminology used herein is for the purpose of describing various embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0048] Any reference to an element or a feature herein using a designation such as “first”, “second” and so forth does not limit the quantity or order of such elements or features, unless stated or the context requires otherwise. For example, such designations may be used herein as a convenient way of distinguishing between two or more elements or instances of an element. Thus, a reference to first and second elements does not necessarily mean that only two elements can be employed, or that the first element must precede the second element, unless stated or the context requires otherwise. In addition, a phrase referring to “at least one of” a list of items refers to any single item therein or any combination of two or more items therein.

[0049] In order that the present invention may be readily understood and put into practical effect, various example embodiments of the present invention will be described hereinafter by way of examples only and not limitations. It will be appreciated by a person skilled in the art that the present invention may, however, be embodied in various different forms or configurations and should not be construed as limited to the example embodiments set forth

hereinafter. Rather, these example embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the present invention to those skilled in the art.

[0050] In particular, for better understanding of the present invention and without limitation or loss of generality, various example embodiments of the present invention will now be described with respect to an example practical application of vision-based structural health monitoring (SHM) for building defect detection for illustration purposes only, whereby the neural network model comprising a first neural network portion configured to perform image classification (e.g., multi-attribute multi-task classification) and a second neural network portion configured to perform building defect localization and/or segmentation. It will be understood by a person skilled in the art that the present invention is not limited to such an example practical application and may be employed in a variety of other practical applications as desired or as appropriate without going beyond the scope of the present invention, as long as image classification and image localization and/or segmentation are required, such as but not limited to, medical image classification and medical image localization and/or segmentation, agricultural image classification (e.g., identifying different plants) and agricultural image localization and/or segmentation, and so on.

[0051] Nowadays, many new technologies have been developed to monitor human health conditions, e.g., using wearable devices to track the heart rate and perform self-diagnosis, which significantly benefits from the boosting of machine learning (ML) and deep learning (DL) techniques based on a large amount of data. Similar to humans, building and bridge structures also have their own health conditions, described by the damage state, the severity, etc., and the structural health condition recognition procedure may be referred to as structural health monitoring (SHM). Knowing the health condition of a structure is useful for various types of decision making, such as whether to repair the structure when cracks are observed or whether to release immediate evacuation signals after catastrophic disasters, e.g., a major earthquake. With rapid developments in sensor hardware and data collection, data-driven SHM has become one of the most active research areas in structural engineering, especially vision-based SHM using image or video data. Analogous to general computer vision (CV) problems (image classification, object detection/localization and object segmentation), image classification is the basic application in vision-based SHM.

[0052] In recent years, there is an increasing trend in using machine learning / deep learning in vision-based SHM, which resulted in a significant performance improvement over traditional

computer vision methods, e.g., edge detection based on extracted vision features. However, many studies are only concerned with the existence of structural damage in the images, and simply treat the problem as single-attribute classification where each labelled image only has one label, i.e., damaged or undamaged. Actually, vision patterns in structural images provide abundant information far more than only the damage state, e.g., scale of the object, type of structural component, severity of damage, etc., which is similar to a multi-attribute recognition problem, and these multiple attributes can be more informative for various types of decision making, such as rapid post-disaster decision making. Moreover, these multiple attributes may have hidden/intrinsic relationships with each other, e.g., hierarchy. For example, the state of a wall being heavily damaged is positively related to the state of the wall being damaged. Therefore, various example embodiments rethink the demands of vision-based SHM and approach the vision-based SHM as a multi-attribute classification problem considering inter-attribute relationships. Furthermore, based on domain expertise, there may be prior knowledge on attributes of direct interest to the SHM, e.g., compared to knowing the color of a building, whether the building has collapsed is more important to SHM. In various example embodiments, for each of the specific attributes, the most correct label is applied to describe the attribute among a number of choices, e.g., the attribute of a structural component type may have one of a number of possible labels such as “beam”, “column”, “wall”, etc. Therefore, each attribute can be treated as a single classification task and an image with multiple attributes may then be subjected to multiple classification tasks, thereby forming a multi-task classification problem.

[0053] In general computer vision, there are many open-sourced datasets for validation of new algorithms and models, e.g., the MNIST (Modified National Institute of Standards and Technology) database, the CIFAR-10 (Canadian Institute for Advanced Research, 10 classes) and the ImageNet database. Compared to these datasets, it is more challenging to deal with structural image dataset for a number of reasons. Firstly, structural images are recognized in a more abstract way. For example, in the CIFAR-10 dataset, contents in the images are usually natural objects, e.g., cat, dog, ship, etc., whose vision patterns are regular and easy to be understood. But in structural images, the descriptions of certain attributes, e.g., damage level of a column, are based on the subjective judgement of experts according to their experience, and thus such vision patterns may not be uniformly/consistently described. Furthermore, labeling structural images requires specific domain knowledge, which increases the cost and difficulties in obtaining a large-scale dataset. For example, this is similar to the challenges met in building medical image dataset. Compared to various types of multi-attribute datasets, e.g.,

AwA (Animals with Attributes) dataset, Clothing Attributes Dataset, CelebA (CelebFaces Attributes) dataset, etc., structural images are more variable, have irregular patterns and noise. For example, the debris of a house after an earthquake presents irregular and chaotic vision patterns, and vehicles in the images are unrelated to building damage detection. Due to complex vision patterns in the images, missing or wrong label in multi-attribute images can also easily occur in annotation procedure. Thus, building such a structural image dataset (multi-attribute multi-task structural image dataset) is not only beneficial to structural engineers, but also contributes to general computer vision to explore the state-of-the-art techniques in a real engineering testbed.

[0054] Accordingly, various example embodiments advantageously employ multi-attribute multi-task classification in the vision-based SHM and obtain a hierarchical multi-attribute multi-task structural image dataset.

[0055] Based on engineering demands and past experience, Gao, Y. and Mosalam, K. M. (2018), "Deep transfer learning for image-based structural damage recognition", *Computer-Aided Civil and Infrastructure Engineering* 33(9):748-768, proposed a general structural image detection framework referred to as ϕ-Net, which included several basic classification tasks for the purpose of automated damage assessment. According to the tree framework in ϕ-Net, a target image is processed following the tree branch layer by layer. Extending from this ϕ-Net, various example embodiments extract eight key structural attributes, namely, (1) scene level, (2) damage state, (3) spalling condition, (4) material type, (5) collapse mode, (6) component type, (7) damage level, and (8) damage type, and then design example classification tasks as example benchmark problems accordingly. Furthermore, various example embodiments re-organize them into a structural image detection framework referred to as ϕ-NeXt (e.g., Next version of ϕ-Net) and shown in FIG. 4. It will be appreciated by a person skilled in the art that these example structural attributes and example classification tasks are provided for illustration purpose only and the present invention is not limited thereto. In particular, it will be appreciated that any types of structural attributes and any types of classification tasks may be provided as long as they are suitable or appropriate for the particular practical application of interest. For illustration purpose, the above example benchmark problems will be investigated later below via benchmark experiments and may be defined as follows:

- Classification Task 1 - Scene Level (3-class classification): *Pixel level*, *Object level* and *Structural level*, which denote how close to a structure (or a part/component or portion thereof) in an image the image is taken from, namely, a close range, a mid-range and a

far range, respectively.

- Classification Task 2 - Damage State (binary classification): *Damaged* and *Undamaged*, which denote whether a structure (or a part/component or portion thereof) in an image is at a damaged state or an undamaged state, respectively.
- Classification Task 3 - Spalling Condition (binary classification): *Spalling* and *Non-spalling*, which denote whether a structure (or a part/component or portion thereof) in an image is at a spalling or non-spalling state, respectively. For example, spalling may refer to the loss of cover material from a structural component surface.
- Classification Task 4 - Material Type (binary classification): *Steel* and *Others*, which denote whether the material of a target structure (or a part or portion thereof) in an image is steel or not steel (i.e., material other than steel), respectively.
- Classification Task 5 - Collapse Mode (3-class classification): *Non-collapse*, *Partial collapse* and *Global collapse*, which denote the severity of damage (three severity/intensity levels) of a structure (e.g., a building) in an image. This classification task may only be performed if the scene level of the image is detected as structural level.
- Classification Task 6 - Component Type (4-class classification): *Beam*, *Column*, *Wall* and *Others*, which denote the type of structural component in an image among four categories. This classification task may only be performed if the scene level of the image is detected as object level.
- Classification Task 7 – Damage Level (4-class classification): *Undamaged*, *Minor damage*, *Moderate damage* and *Heavy damage*, which denote the severity of damage (four severity/intensity levels) of a structural component in an image. This classification task may only be performed if the scene level of the image is detected as object level.
- Classification Task 8 - Damage Type (4-class classification): *Undamaged*, *Flexural damage*, *Shear damage* and *Combined damage*, which denote the type of damage on a structural component in an image. This classification task may only be performed if the scene level of the image is detected as object level. For example, such an object level image may have a complex, irregular and even abstract semantic vision pattern.

[0056] Based on hierarchical relationships and inter-task (attribute) dependency in ϕ-NeXt as shown in FIG. 4, one structural image may go through multiple classification tasks where the output (label) of each classification task is treated as a single structural attribute. Therefore, a sequential set of attributes can be obtained for structural health assessment. For illustration purpose, FIG. 5 shows a number of example multi-attribute structural images, along with the

attributes associated with each structural image. For each classification task, the metric used is accuracy, defined as the ratio between the number of correct predictions to the total number of predictions. The objective of each classification task is to gain as high an accuracy as possible. In addition, according to domain knowledge, classification tasks 1 to 4 may be considered as easy (or easier) tasks or problems, where people with or without relevant background are also able to easily select the correct label using the above definitions for these classification tasks. On the other hand, classification tasks 5 to 8 may be considered as hard (or harder) tasks or problems, especially for classification task 7 and 8 even for experts.

ϕ-NeXt Dataset Establishment

Data Collection and Labeling

[0057] From multiple resources (e.g., online structural engineering database Seismic Performance Observatory, search engine Google Image, personal donation, etc.), nearly 100,000 raw images were collected. To pursue better image qualities, before labeling, preprocessing was briefly conducted to filter out low-resolution images (e.g., lower than 224×224 pixels) and noisy images with too many irrelevant contents. To obtain labels of multiple attributes, labeling work was crowdsourced and performed via an online labeling tool, such as shown in FIG. 6, whereby the labeling sequence for each attribute may be based on the original ϕ-Net. About 20 volunteers with specific SHM background were engaged for this labeling exercise. To avoid biased labels, majority voting was adopted. Accordingly, images were labeled by multiple volunteers, only images with all associated attributes having at least 3 votes were adopted, and the majority vote decided the label for each attribute. Overall, 36,413 images with multiple labels (i.e., 36,413 multi-attribute labelled images) were obtained.

Training and Test Splitting

[0058] For the benchmarking purpose of training and validating machine learning and deep learning models, fixed training and testing datasets were split and open-sourced, where a roughly 8:1 ratio was adopted. Unlike single-attribute classification tasks, a coarse and random split may disrupt the distributions of labels in training and test datasets and lead to some extreme cases, e.g., all labels for the collapse mode attribute (Task 5) are in the training dataset, and images in the test dataset have no labels for the collapse mode attribute. After multiple iterations, 32,407 training images and 4,006 test images were obtained. For each attribute, the ratios between training labels and test labels were all close to 8:1 or 9:1 and the label distributions

were also identical between training and test datasets. FIG. 7 depicts a bar graph showing the statistics of training and test label distributions for classification tasks 1 to 8. In particular, FIG. 7 shows that:

- for classification task 1, there were 7,690, 8,111 and 8,508 training labels for pixel level, object level and structural level attributes, respectively, and 965, 962 and 1,070 test labels for pixel level, object level and structural level attributes, respectively;
- for classification task 2, there were 6,282 and 5,529 training labels for undamaged state and damaged state attributes, respectively, and 745 and 715 test labels for undamaged state and damaged state attributes, respectively;
- for classification task 3, there were 2,604 and 4,294 training labels for non-spalling condition and spalling condition attributes, respectively, and 310 and 527 test labels for non-spalling condition and spalling condition attributes, respectively;
- for classification task 4, there were 1,806 and 6,506 training labels for steel and non-steel (i.e., others) attributes, respectively, and 209 and 770 test labels for steel and non-steel (i.e., others) attributes, respectively;
- for classification task 5, there were 322, 379 and 525 training labels for non-collapse state, partial collapse state and global collapse state, respectively, and 39, 40 and 67 test labels for non-collapse state, partial collapse state and global collapse state, respectively;
- for classification task 6, there were 511, 1,618, 2,268 and 358 training labels for beam, column, wall and others attributes, respectively, and 60, 205, 265 and 49 test labels for beam, column, wall and others attributes, respectively;
- for classification task 7, there were 1,551, 869, 799 and 919 training labels for undamaged state, minor damage state, moderate damage state and heavy damage state attributes, respectively, and 207, 93, 104 and 94 test labels for undamaged state, minor damage state, moderate damage state and heavy damage state attributes, respectively; and
- for classification task 8, there were 1,598, 476, 826 and 1,193 training labels for undamaged state, flexural damage type, shear damage type and combined damage type attributes, respectively, and 215, 46, 99 and 132 test labels for undamaged state, flexural damage state, shear damage state and combined damage state attributes.

Dataset Extensions

[0059] As mentioned above, this engineering-based multi-attribute dataset can not only serve as a benchmark dataset in structural engineering, but also can contribute to general computer vision / machine learning / deep learning fields in multi-attribute or multi-task studies. Due to various uncertainties in the labeling procedure (therefore, volunteers bypassed certain images or attributes and certain images were excluded due to the three votes needed for validity), it was not possible to obtain complete attributes, which explains the observation in FIG. 7 that, for each classification task, the total number of labels obtained was less than the total number of possible labels for the classification task. Thus, this situation may introduce a missing label problem and this multi-attribute dataset can provide a new testbed for relevant studies. Moreover, the multi-attribute dataset can be further separated into eight task-oriented subsets for single-attribute classification, and these subsets may also introduce certain challenges. For example, compared to classification tasks 1 to 4, classification tasks 5 to 8 have very limited labelled data, especially in classification task 5, thereby forming a small-scale dataset problem. Moreover, in classification task 4, to identify the material type, the label "others" is significantly over-represented than the label "steel", which is consistent to real life that steel material is only a small portion of the entire material family, thereby forming an unbalanced labeling problem. Therefore, the ϕ-NeXt dataset can be treated as a new benchmark dataset for either multi-attribute or multi-task, and its single-attribute subsets can also be applied to small-scale or unbalanced dataset problem.

Example Methods and Evaluations

[0060] In order to set the benchmark performance on the ϕ-NeXt dataset, a series of experiments was conducted and the influence of different approaches/methods and models in classification accuracy improvement were compared. The experiments were conducted on TensorFlow and Keras platform and performed on CyberpowerPC with single GPU (CPU: Intel Core i7-8700K@3.7GHz 6 Core, RAM:32GB and GPU: Nvidia Geforce RTX 2080Ti).

Baseline Networks

[0061] Firstly, the eight classification tasks were treated independently and each time only the accuracy of a single classification task was evaluated. Three classic deep convolutional neuronal network (CNN) models were adopted, namely, VGG16, VGG19 and ResNet50. They were trained from scratch and may be denoted as baseline (BSL). As a common method to

enhance model performance (especially for small dataset), data augmentation via a random combination from shifting, zooming, etc. was applied, which may be denoted as BSL-DA, and compared to baseline. For simplicity, one training run was performed for most cases except for some cases where the loss does not change at the first 10 epochs.

[0062] For a fair comparison, the experiment settings for each classification task and model were the same. Batch size was 64 and maximal training epoch was 50. The stochastic gradient descent with piece-wise decayed learning rate was adopted where the learning rate was divided by 10 when test accuracy was trapped at a plateau. Besides dropout, no extra regularizers were applied and dropout rate was 0.5. The only difference compared to prototype VGG and ResNet architecture was that the number of neurons in the last output layer was modified to adapt to the number of classes in different classification tasks. All three models were trained from scratch, where model parameters were initialized from random space. While conducting data augmentation, training images in each batch were transformed with a random combination of the following 6 example augmentation options: (1) horizontal translation within 10% of total width, (2) vertical translation within 10% of total height, (3) rotation within 5°, (4) zoom in less than 120% of original size, (5) zoom out less than 80% of original size, and (6) horizontal flip.

[0063] The test accuracies of the three CNN models under the baseline (baseline results (%)) with and without data augmentation (BSL-DA and BSL, respectively) are shown in Table 1 in FIG. 8A. In general, ResNet50 produced a better performance than VGGNets with higher test accuracies, especially in hard classification tasks, namely, 13%, 17% and 20% enhancement in classification tasks 5, 7 and 8, respectively. Under the baseline setting, data augmentation did not achieve significant improvement, e.g., 1% to 2% in most cases and even sometimes produced slightly worse performance. This can be partially addressed by the fair setting in data augmentation for all cases. As a result, task-orientated and more elaborate augmentation settings can be expected in future studies. Further important observations not presented in Table 1 are discussed below. In all cases, training accuracies were higher than test but far lower than 100%. Difference between training and test was observed more obviously in ResNet50 cases where a difference of 10% to 20% occurred in classification tasks 5 to 8, but for VGGNets, the difference was small and within 6%. From the training histories, curves of VGG16 and VGG19 were flat from the start of training in classification tasks 3, 5, 7 and 8, which indicates an un-trainable situation. Even though multiple runs were conducted, this observation was consistent. On the contrary, ResNet was still trainable, which indicates a better generalization in the classification tasks under the baseline setting. Nevertheless, from the point of view of practical

considerations, the values of accuracy obtained for the baseline are far from satisfactory, and thus, further enhancements may be desired.

Transfer Learning Approach

[0064] In deep learning studies, transfer learning (TL) is a technique which may be employed to improve performance by transferring knowledge from one source domain to another target domain, which has been demonstrated to be effective on some small dataset problems (e.g., medical image). Compared to ImageNet, the scale of ϕ-NeXt is far smaller, various example embodiments note that these two datasets still have some overlaps, e.g., ImageNet includes some images relating to structural engineering (e.g., building, pillar, church, etc). For example, even though all of such images fall under the classification of undamaged, they may still contribute to feature learning relating to shape, texture, etc. in the above-mentioned classification tasks. Moreover, pre-trained model from ImageNet is found to have strong generalization on custom dataset. Therefore, various example embodiments adopt transfer learning as one enhancement approach over the baseline, and investigate its performance within the above-mentioned three CNN models, whereby similarly, the classification tasks are treated independently.

[0065] The pre-trained VGG16, VGG19 and ResNet50 from ImageNet were accessed in Keras2. Similar to the baseline, only the last output layer of the CNN models was modified according to number of classes in different classification tasks. No layers were fixed and all layers and parameters were fine-tuned in the training phase. Moreover, learning rate schedule and dropout rate were kept as the same as the baseline. Similar to the baseline, data augmentation was also performed for each model to further explore its influence on performance under the transfer learning setting.

[0066] The test accuracies of the three CNN models under the transfer learning approach (transfer learning results (%)) with and without data augmentation (TL-DA and TL, respectively) are summarized in Table 2 shown in FIG. 8B. It can be observed that, under the transfer learning approach, all of the three CNN models achieved obvious improvement over the baseline. For example, it is noted that the un-trainable situations of VGGNets in the baseline have been eliminated, and VGG19 can even obtain competitive performance with ResNet in hard tasks (e.g., the best model for classification tasks 6 and 8 is VGG19). Moreover, in hard tasks, the influence of transfer learning is found to be more significant than in easy tasks, where 10% to 20% upstreaming performance above the baseline were obtained. On the other hand,

the influence of data augmentation still appears to be limited under the transfer learning approach, and even produced slight accuracy decrease in some cases. From the training histories, the training accuracies of all cases arrive close to 100%, which indicates that such level of model complexities is able to handle these classification tasks, and slight model modifications or elaborate training techniques with fine-tuned parameters may help to further enhance the accuracy. From the point of view of practical considerations, test accuracies presented for classification tasks 1 and 4 may be acceptable in real applications, but for the remaining classification tasks, especially the hard tasks, further improvement may be desired.

Hierarchical Transfer Learning (HTL) Approach

[0067] In the above two experiments (baseline and transfer learning experiments), each classification task (or attribute) was treated as independent and the influence between classification tasks were not taken into account. However, based on domain knowledge, various example embodiments note that these classification tasks have certain relationships between each other, especially the hierarchical structure shown in FIG. 4. Various example embodiments already utilized such a hierarchical relationship in building the ϕ-NeXt dataset as described hereinbefore. In this regard, various example embodiments further utilize this hierarchical relationship to investigate whether domain expertise can help to enhance accuracy. The above transfer learning experiments demonstrated a great amount of knowledge transfer from ImageNet to the above-mentioned classification tasks, and a significant improvement was achieved compared to the baseline. In various example embodiments, experiments were conducted using the transfer learning approach as well as the knowledge between classification tasks transferred through certain or predetermined hierarchical paths (e.g., certain or predetermined paths defined in the ϕ-NeXt framework shown in FIG. 4). Thus, such an approach may be referred to herein as the hierarchical transfer learning (HTL) approach, which seeks to improve model performance, especially in hard tasks, based on knowledge and information from easy tasks.

[0068] For benchmarking purpose, in the hierarchical transfer learning experiment, four transfer paths were designed based classification tasks 1, 2, 3, 7 and 8, which may be referred to herein as a simplified ϕ-NeXt framework. Accordingly, different attributes are linked together via the simplified ϕ-NeXt framework. The experiment seeks to demonstrate that hard tasks 7 and 8 can benefit from knowledge and information from easy tasks 1 and 2. For example, classification task 1 scene level has the most images compared to other classification tasks, and

thus, depending on hierarchical relationships, classification task 1 can provide additional knowledge and information for subsequent classification tasks in the hierarchical structure. For example, the damage state (associated with classification task 2) may be the most important attribute in the vision-based SHM, and as can be seen in the hierarchical structure shown in FIG. 4, it has a direct relationship with the spalling condition (associated with classification task 3), the damage level (associated with classification task 7) and the damage type (associated with classification task 8). For example, if one structural image has no damage patterns, various example embodiments note that the label 'Undamaged' in classification task 2 is strongly and positively related to the label 'Non-spalling' in classification task 3, as well as the label 'Undamaged' in classification tasks 7 and 8, and so on. Thus, various example embodiments find that labels of the scene level and the damage state may help the model better understand hard tasks (e.g., classification tasks 7 and 8).

[0069] The above-mentioned four paths (simplified hierarchical transfer learning paths) are illustrated in FIG. 9. Path A follows the ϕ-NeXt framework but removed relevant classification task 4 (associated with the material type), which is general and has already achieved promising results, and other irrelevant tasks due to the constraints that classification tasks 3, 7 and 8 are subsequent classification tasks if the image is classified to be at the object level. Path B generally follows path A but ignores classification task 2 associated with the damage state. On the other hand, path C utilizes the damage state to replace the scene level. Path D does not follow the hierarchical structure shown in FIG. 9 but explores the relationships between several subsequent tasks of object level images. For example, if one object level image is classified as 'Spalling' in classification task 3, it is likely related to label 'Heavy damage' in classification task 7 and 'Combined damage' in classification task 8. This is because if the spalling area is large, i.e., a large portion of cover material is lost from the surface of the structural component, it is likely that the structural component is heavily damaged and its cause is complex due to combined effects of multiple external forces, which thus relates to label 'Heavy damage' in classification task 7 and 'Combined damage' in classification task 8.

[0070] For simplicity and clarity, only ResNet50 without data augmentation was employed in the hierarchical transfer learning experiment and the remaining settings were the same as the transfer learning approach. Since the transfer learning approach gained significant improvement over the baseline, the ImageNet pre-trained model was assigned as the first model in each path. Along each path, the model for a current classification task is inherited from the previous classification task and trained (or fine-tuned), and then it is used as a pre-trained model for the

next classification task, and so on. For example, for the route from classification task 1 (first or source classification task) to classification task 2 and then to classification task 7 (target classification task), the ImageNet pre-trained model is employed for the classification task 1 and trained (or fine-tuned), the trained model for the classification task 1 is then employed as a pre-trained model for the classification task 2 and trained (or fine-tuned), and the trained model for the classification task 2 is in turn employed as a pre-trained model for classification task 7 and trained (or fine-tuned). Without loss of generality, at least three training runs were performed for each classification task and the best model was then selected.

[0071] The test accuracies of the ResNet50 without data augmentation (hierarchical transfer learning results (%)) under the HTL approach are presented in Table 3 shown in FIG. 10. Compared to the transfer learning results without data augment, in general, classification task 8 can gain nearly 1% to 2% enhancement in all cases, but there exists a slight accuracy decrease in classification task 7 regardless of the path chosen. Compared amongst between the four paths, for the purpose of improving hard tasks, path A is considered as the best, which has the highest test accuracy in classification tasks 7 and 8. In addition, it was found that classification tasks 2 and 3, as intermediate tasks, also benefit from the hierarchical relationships, which gained 1% and 2% performance improvements, respectively. It can also be observed that in paths B and C, with pre-training from classification task 1 or 2, task 3 can achieve 1.4% performance improvement. Compared to transfer learning using data augmentation, HTL shares similar task-dependent characteristics, where it can either obtain small improvement or render slightly worse results in some cases. But interestingly, while adopting data augmentation in classification task 8, the accuracy decreased by 0.8%; on the contrary, 2.0% enhancement was obtained under HTL. Accordingly, for classification task 8, taking into account hierarchy provides a better performance than normal approach.

[0072] Accordingly, using hierarchical relationship based on domain expertise produces improvements in some cases, but may still have certain drawbacks. In the HTL approach, it is assumed that partial data in previous classification tasks can contribute to subsequent classification tasks, e.g., more object level images with damaged or undamaged state stored in classification tasks 1 and 2 datasets may provide additional information for spalling condition, damage level and damage type recognition. However, various example embodiments note that this information is parametrized by deep learning models, where it is hard to be understood and easy to be influenced. For example, suppose the source classification task is classification task 1 and target classification task is classification task 7, the large amount of structural level

images in classification task 1 may transfer some knowledge related to the collapse mode, which describes the damage severity at the structural level. However, the objective of classification task 7 is to recognize the damage severity level at the object level. Thus, the parametrized information transferred from classification task 1 to classification task 7 may confuse the classifier in classification task 7. However, this point cannot explain the improvement observed in the HTL approach in classification task 8 which also deals with the specific problem of object level images. Thus, the HTL approach may not fully capture and exploit the inter-task relationship. Some related work adopted recurrent neural networks (RNN) and long short-term memory networks (LSTM) to capture the relationship between multiple inputs, which may be evaluated in future studies. As for computational cost, performing HTL does not introduce extra computation compared to the transfer learning approach, because it just fine-tunes the model parameters based on the previous classification task instead of ImageNet. However, there are similar issues with the baseline and transfer learning, such as the need to perform eight times respectively for the eight structural attribute classification tasks, even though the classification tasks are no longer independent in HTL. To address this problem, various example embodiments provide an approach whereby the network model is configured to train all classification tasks together (as a whole) while taking into account inter-relationship, that is, multi-task learning, which will now be described below.

Multi-Task Learning (MTL) Approach for Classification

[0073] To further take advantage of the inter-relationships among classification tasks, various example embodiments employ multi-task learning (MTL). MTL seeks to learn to share the representations among multiple different classification tasks, which can enable the model to generalize well for these classification tasks. For example, in the ϕ-NeXt dataset, there exist some connections or inter-relationship among the eight classification tasks. For instance, classification tasks 7 and 8 aim to classify the damage level and type, respectively, and due to the similarity in their objectives, they may contribute to the feature learning of each other. Even though they may have large amounts of dataset overlap for these classification tasks, multi-attributes may also benefit the generalization ability. To this end, various example embodiments provide a MTL framework using a shared feature extractor for the multiple classification tasks (all the classification tasks) and multiple independent classifiers (for the multiple classification tasks) which may be multiple fully connected layers. In various example embodiments, the shared feature extractor may be any feature extractor known in the art as desired or as

appropriate, and preferably is a state-of-the-art feature extractor for computer vision tasks, which for example currently is the Swin Transformer.

[0074] FIG. 11 depicts a schematic drawing of a MTL framework or model 1100 according to various example embodiments of the present invention. In particular, the MTL model 1100 comprises a shared feature extractor 1104 and a plurality of task classifiers 1108 associated with a plurality of attributes, respectively, connected to the shared feature extractor 1104. To capture the effective features among all classification tasks in an end-to-end manner, the MTL model 1100 is trained simultaneously using the whole ϕ-NeXt dataset where each image sample (labelled image) may have multiple labels (related to multiple attributes). For example, for a batch of image samples X in an iteration, various example embodiments update the parameters of the shared feature extractor 1104 and the corresponding classifiers 1108 by its labels using cross-entropy loss, as shown in FIG. 11. In this way, once the MTL model 1100 is trained for one iteration, the whole model 1100 is updated once for all of the plurality of classification tasks 1108 (e.g., the above-mentioned eight classification tasks) accordingly, and thus advantageously avoids the need to train the model multiple times for multiple classification tasks (e.g., eight times for the eight classification tasks), respectively.

[0075] Similar to the above-mentioned HTL settings, for simplicity and clarity, only ResNet50 was adopted for the MTL model in experiments conducted, including a Swin Transformer as the backbone, and the ImageNet pre-trained parameters were utilized before training. The eight classifiers are each configured as a fully connected layer having an output dimension configured according to the number of classes associated with the classification task. All the training strategies were the same or similar as the baseline. For a fair comparison, in addition to conducting an experiment on the MTL model without data augmentation (denoted as MTL), an experiment was also conducted on the MTL model with data augmentation (denoted as MTL-DA).

[0076] The test accuracies of the MTL (MTL results (%)) are shown in Table 4 as shown in FIG. 12. Compared to both transfer learning (TL) and HTL results based on ResNet50, clear improvement can be observed for the MTL approach for most classification tasks regardless of whether data augmentation was adopted or not, which indicates the state-of-the-art performance over the baseline. For example, especially for the MTL on classification task 6 and the MTL-DA on classification task 8, significant improvements over the transfer learning model by 4.2% and 5.9%, respectively, were observed. The only exception was classification task 5, which is the collapse mode analysis task, for which the performance of the MTL model decreased by

2.8% without data augmentation. Such a drop may be explained based on domain knowledge as follows. Classification task 5 (collapse mode) is the only subsequent task to structural level images, which is quite irrelevant to other classification tasks based on pixel level or object level images. For example, knowing the material type (classification task 4) does not provide related information to identify whether the building collapsed and knowing how severely the column is damaged (classification task 7) may even confuse the classifier for detecting the collapse mode, because column and building are from different scales and scene levels. Therefore, there is very little overlap between classification task 5 and other classification tasks in the ϕ-NeXt framework. Accordingly, without common images and similar/related classification tasks, training the MTL model on many pixel-level and object-level images may hinder the performance of classification task 5.

[0077] Accordingly, from the perspective of vision-based SHM, the MTL approach has several advantageous, such as: (1) it achieves the state-of-the-art performance in all classification tasks except classification task 5, (2) it exploits the inter-relationships between classification tasks, and (3) it enhances training efficiency significantly compared to other approaches described hereinbefore.

Multi-task Heterogeneous Learning Framework

[0078] Various approaches for classifying images (e.g., building defect images) have been described hereinbefore according to various example embodiments, including the MTL approach. In this regard, various example embodiments note that the image localization and/or segmentation (e.g., localization and/or segmentation of building defect(s) in a structural image) also have important practical applications. In particular, there is a wide range of practical applications in which image classification and image localization and/or segmentation may be desired or required, including the example practical application of vision-based SHM for building defect detection, including building defect classification and building defect localization and/or segmentation. In this regard, various example embodiments note that conventional approaches are to train separate neural network models for image classification tasks and image localization and/or segmentation tasks. However, various embodiments note that there are a number of drawbacks associated with such conventional approaches, including being inefficient (e.g., the need to train these neural network models separately, which wastes resources), being inconvenient/cumbersome (e.g., multiple separate neural network models need to be deployed to perform the different types of tasks) and providing unsatisfactory

performances (e.g., relationships/correlations between different types of tasks associated with a practical application not being captured (e.g., does not leverage the correlation between classification and localization/segmentation tasks) and performance degradations when deployed in practical applications compared with those obtained based on training data). To overcome or address one or more of these problems, various example embodiments provide a multi-task heterogeneous learning method or framework for image classification (e.g., multi-task classification such as described hereinbefore according to various example embodiments) and image localization and/or segmentation (e.g., building defect localization and/or segmentation in an image).

[0079] Accordingly, various example embodiments combine multi-task learning (e.g., the MTL approach for classification as described hereinbefore according to various example embodiments) with heterogeneous learning (training a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or segmentation as one neural network model (as a whole)) to advantageously enable the neural network model to concurrently handle image classification tasks and image localization and/or segmentation tasks. In various example embodiments, the heterogeneous learning comprises training the first and second neural network portions based on a backbone sharing technique (e.g., weight sharing technique) which will be described in more detail later below.

[0080] FIG. 13 depicts a schematic drawing of the multi-task heterogeneous learning framework or model 1300 according to various example embodiments of the present invention, such as corresponding to the neural network model subjected to the training method 100 as described hereinbefore according to various embodiments of the present invention. The model 1300 comprises a first neural network portion 1310 (which may be referred to herein as neural network portion or part 'a', or simply as portion or part 'a') configured to perform image classification and a second neural network portion 1320 (which may be referred to herein as neural network portion or part 'b', or simply as portion or part 'b') configured to perform image localization and/or segmentation. The first neural network portion 1310 comprises a feature extractor 1312 (which may be referred to as a backbone of the first neural network portion 1310) and a plurality of task classifiers 1314 associated with a plurality of attributes, respectively, connected to the feature extractor 1312. For example, the first neural network portion 1310 may correspond to the MTL model 1100 as described hereinbefore according to various example embodiments whereby the feature extractor 1312 is a shared feature extractor

with respect to the plurality of task classifiers 1314 (e.g., fully connected layers). The second neural network portion 1320 comprises a feature extractor 1322 (which may be referred to as a backbone of the second neural network portion 1310), along with a bounding box predictor 1324 configured to perform image localization and/or a mask predictor 1326 configured to perform image segmentation. In various example embodiments, the feature extractors 1312, 1322 may each be any feature extractor known in the art as desired or as appropriate, and preferably is a state-of-the-art feature extractor for computer vision, which for example currently is the Swin Transformer. In other words, according to various example embodiments, the feature extractors 1312, 1322 are preferably Swin Transformers.

[0081] For example, as a state-of-the-art neural network architecture based on self-attention mechanisms, a Swin Transformer possesses advantages in various aspects that traditional Convolutional Neural Networks (CNNs) may struggle to achieve, such as in long-range dependency modeling, position encoding and multi-scale adaptability. For example, a Swin Transformer excels in modeling long-range dependencies within images, which is a crucial factor for tasks such as capturing global context or intricate relationships between objects in large-sized images. In this regard, traditional convolutional operations may encounter limitations when dealing with long-range dependencies. For example, various example embodiments note that structural defects in housing in Singapore tend to occur in the vicinity of water taps. With CNNs, due to their limited receptive field, the CNN model may struggle to establish a strong or sufficient correlation between the spatial location of the water tap and the defect. However, with the Swin Transformer, the model was found to be able to effectively establish a strong connection between the spatial location of the water tap and the defect, leading to a more accurate identification and localization of targets. In addition, the Swin Transformer employs relative position encoding to accurately represent relationships between different positions, which is especially beneficial for larger images. This aids the model in better comprehending contextual information across various regions of the image. For large-sized images, the Swin Transformer holds a significant advantage over traditional CNN models. For example, the acquisition of images depicting architectural defects may typically be reliant on devices such as digital cameras or smartphones, resulting in images with high resolutions. Therefore, applying a Swin Transformer for image segmentation and localization algorithms yields notably enhanced accuracy compared to CNNs, leading to substantial improvements in performance (e.g., accuracy). Furthermore, the Swin Transformer's architecture adapts more easily to inputs of varying scales, which can provide an advantage in various tasks that involve

multiple scales, such as object detection and semantic segmentation. On the other hand, traditional CNNs may require additional processing for different input scales.

[0082] In various example embodiments, the first neural network portion 1310 and the second neural network portion 1320 are different in their final classifiers. In this regard, the classifiers of the first neural network portion 1310 are configured for multi-task classification, which involves detecting/recognizing various attributes of an input image, such as described hereinbefore according to various example embodiments. For example, as described hereinbefore according to various example embodiments, a total of eight classification tasks may be configured and the first neural network portion 1310 may thus be configured to produce predictions for all of the eight classification tasks (i.e., multi-task classification). On the other hand, the second neural network portion 1320 is configured to perform image localization (e.g., predict defect location(s) by enclosing defect(s) with rectangular bounding box(es)) and/or image segmentation (e.g., defect segmentation by highlighting defect(s) with colored pixels (e.g., applying colored mask(s) over the defect(s))) on an input image. In particular, in the heterogeneous learning according to various example embodiments of the present invention, the first and second neural network portions 1310, 1320 are trained in a manner which facilitates or enables the sharing of their backbone parameters (e.g., weight parameters) therebetween, which has been found to enhance performances of the model 1300 in image classification and image localization and/or segmentation in practical applications.

[0083] For illustration purpose and by way of an example only, FIG. 14 depicts a schematic drawing of an example architecture 1400 of the neural network model 1300. The example architecture 1400 comprises a first neural network portion 1410 configured to perform image classification corresponding to the above-mentioned first neural network portion 1310 and a second neural network portion 1420 configured to perform image localization and segmentation corresponding to the above-mentioned second neural network portion 1320. The first neural network portion 1410 comprises a feature extractor 1412 implemented as a Swin Transformer and a plurality of task classifiers 1414 (e.g., fully connected layers) associated with a plurality of attributes, respectively, connected to the feature extractor 1412. The second neural network portion 1420 comprises a feature extractor 1422 also implemented as a Swin Transformer, along with a bounding box predictor 1424 configured to perform image localization and a mask predictor 1426 configured to perform image segmentation.

Backbone Sharing Technique

[0084] According to various example embodiments, the backbone sharing technique comprises: training, at a first training round, the first neural network portion 1310 based on a training dataset to obtain first backbone parameters of the first neural network portion 1310 corresponding to the first training round; applying, at the first training round, the first backbone parameters (e.g., first weight parameters) of the first neural network portion 1310 obtained corresponding to the first training round to the second neural network portion 1320; and training, at the first training round, the second neural network portion 1320 based on the training dataset to obtain second backbone parameters (e.g., second weight parameters) of the second neural network portion 1320 corresponding to the first training round. The technique further comprises performing, after the first training round, a plurality of additional training rounds. Each additional training round comprises: updating, at the additional training round, the first backbone parameters of the first neural network portion 1310 based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round (i.e., based on the second backbone parameters obtained corresponding to the first training round if there is no previous additional training round with respect to the current additional training round or otherwise based on the second backbone parameters obtained corresponding to the immediately previous additional training round); training, at the additional training round, the first neural network portion 1310 based on the training dataset to obtain the first backbone parameters of the first neural network portion 1310 corresponding to the additional training round; updating, at the additional training round, the second backbone parameters of the second neural network portion 1320 based on the first backbone parameters obtained corresponding to the additional training round; and training, at the additional training round, the second neural network portion 1320 based on the training dataset to obtain the second backbone parameters of the second neural network portion 1320 corresponding to the additional training round.

[0085] In various example embodiments, the above-mentioned updating, at the additional training round, the first backbone parameters of the first neural network portion 1310 is further based on a first backbone sharing parameter. In various embodiments, similarly, the above-mentioned updating, at the additional training round, the second backbone parameters of the second neural network portion 1320 is further based on a second backbone sharing parameter.

[0086] In various example embodiments, the first backbone sharing parameter is configured to control an amount of the second backbone parameters obtained for updating the first

backbone parameters of the first neural network portion 1310. In various example embodiments, similarly, the second backbone sharing parameter is configured to control an amount of the first backbone parameters obtained for updating the second backbone parameters of the second neural network portion 1320.

[0087] In various example embodiments, the above-mentioned performing the plurality of additional training rounds comprises performing a number of additional training rounds until each of the first and second neural network portions 1310, 1320 converges.

[0088] For illustration purpose and by way of an example only, an example implementation of the backbone sharing technique will now be described according to various example embodiments of the present invention.

[0089] Firstly, the first neural network portion 1310 (e.g., network portion 'a' shown in FIG. 13) may be trained based on a training dataset until it converges. The backbone parameters g^a (e.g., weight parameters) of network portion "a" 1310 may then be obtained.

[0090] The backbone parameters of portion "a" 1310 obtained may then be applied (e.g., copied or transferred) to the backbone parameters g^b (e.g., weight parameters) of the second neural network portion 1320 (e.g., network portion 'b' shown in FIG. 13). The network portion 'b' 1320 may be trained based on the training dataset for one epoch and the backbone parameters g^b of network portion "b" 1320 may then be obtained. It will be appreciated by a person skilled in the art that the number of training epoch performed is not limited to one epoch and may be varied depending on various factors, such as the size of the training dataset. For example, the number of training epochs performed may range from 1 to 3.

[0091] The above operations or steps of the example implementation described so far may correspond to operations or steps described above under the first training round. The operations or steps of the example implementation described below may correspond to operations or steps described above under the plurality of additional training rounds.

[0092] The backbone parameters g^a may then be updated based on the backbone parameters g^b obtained as follows:

$$g^a = \lambda_1 g^a + (1 - \lambda_1) g^b$$

(Equation 1)

[0093] The network portion 'a' 1310 (with the updated backbone parameters g^a) may then be trained based on the training dataset for one epoch and the backbone parameters g^a may then be obtained. As explained above, it will be appreciated by a person skilled in the art that the

number of training epoch performed is not limited to one epoch and, for example, may range from 1 to 3.

[0094] The backbone parameters g^b may then be updated based on the backbone parameters g^a obtained as follows:

$$g^b = \lambda_2 g^b + (1 - \lambda_2) g^a$$

(Equation 2)

[0095] The network portion 'b' 1320 (with the updated backbone parameters g^b) may then be trained based on the training dataset for one epoch and the backbone parameters g^b may then be obtained.

[0096] The above-described updating of the backbone parameters g^a and g^b and the training of the network portions 'a' and 'b' 1310, 1320 may continue iteratively until each of the network portions 'a' and 'b' 1310, 1320 converges.

[0097] The parameters λ_1 and λ_2 correspond to the above-mentioned first and second backbone sharing parameters, respectively. As can be seen from Equation (1) above, λ_1 is configured to control an amount of the second backbone parameters g^b obtained for updating the first backbone parameters g^a of the network portion 'a' 1310. Similarly, as can be seen from Equation (2) above, λ_2 is configured to control an amount of the first backbone parameters g^a obtained for updating the second backbone parameters g^b of the network portion 'b' 1320. In this regard, λ_1 and λ_2 may be hyperparameters and may be configured to as appropriate based on various factors, such as depending on the specific tasks. In various example embodiments, the values of λ_1 and λ_2 are set close to 1 to prevent non-convergence during training, and preferably λ_1, $\lambda_2 \in [0.9, 0.99]$, whereby $\lambda_1 \leq \lambda_2$. For example, due to the increased complexity of segmentation tasks, the value of λ_2 may be set slightly greater than λ_1. By way of an example only and without limitation, λ_1 may be set to 0.95 and λ_2 may be set to 0.99. As an example illustration, assuming that $g^a = [1, 2, 3, 4, 5]$, $g^b = [1, 4, 0, 5, 7]$ and λ_1 is set to 0.95, according to Equation (1) above, backbone parameters g^a may be updated based on the backbone parameters g^b as follows:

$$g^a = 0.95 g^a + 0.05 g^b$$

$$g_0^a = 1 * 0.95 + 1 * 0.05$$

$$g_1^a = 2 * 0.95 + 4 * 0.05$$

$$\ldots$$

$$g_4^a = 5 * 0.95 + 7 * 0.05$$

As a result, the updated backbone parameters g^a obtained is: $g^a = [1, 2.1, 2.85, 4.05, 5.1]$.

[0098] By employing the above-mentioned backbone sharing technique, various example embodiments advantageously achieve the framework of multi-task heterogeneous learning. For example, both multi-classification tasks and localization and/or segmentation tasks on an input image can be carried out simultaneously, without the need for laborious training and deployment of multiple separate models, which significantly enhances practicality and convenience, and thus overall usability in practical applications.

[0099] To demonstrate the effectiveness of the heterogeneous learning (HL) using the backbone sharing technique described above according to various example embodiments of the present invention, experiments were conducted using the network portion 'b' 1320 trained based on the backbone sharing technique and using a comparative model trained solely using Mask R-CNN (denoted as MR-CNN, which is a classical image segmentation and localization network). The localization and segmentation results are presented in Table 5 shown in FIG. 15. The evaluation metric in Table 5 is the Average Precision (AP). It can be observed that the heterogeneous learning-based model according to various example embodiments, through its backbone sharing approach (e.g., weight sharing approach) that learns the correlation between classification and localization/segmentation tasks, outperforms the Mask R-CNN model significantly in both localization and segmentation tasks. Additionally, the heterogeneous learning-based model advantageously avoids the need to train and deploy multiple separate neural network models to perform the different types of tasks.

Environment Adaptive Learning/Optimization Technique

[00100] After the model 1300 has been trained using the above-described backbone sharing technique, a trained model θ_{tr} is obtained including the backbone parameters g^a and g^b. In various example embodiments, this trained model θ_{tr} is then fine-tuned to obtain an environment-adaptive model θ_{te}.

[00101] Various example embodiments note that an inevitable challenge when applying deep models in practical scenarios is the problem of model generalization. For example, even if deep models achieve high scores on the training data, deploying these classification and/or localization/segmentation models to new scenarios can lead to a significant decrease in performance. This drop in performance occurs because the data obtained in new scenarios/environments differs from the training data, due to factors such as lighting conditions, architectural styles and so on. For example, to demonstrate the issue associated with different architectural styles, an experiment was conducted whereby two datasets were collected. One

dataset was the well-known ϕ-Net dataset in the civil engineering domain, which included over 36,000 images related to building defects. The other dataset was obtained through the collection of building defect images from residential buildings in Singapore, which is referred to herein as the Residential dataset and included about 3,000 images. An example simple task of identifying whether spalling is present in an image was configured for the experiment, which is a binary classification task. Both datasets focused on building defects, with the key difference being that the ϕ-Net dataset was downloaded from the internet, whereas the Residential dataset was obtained from residential buildings in Singapore.

[00102] The performances of (1) a first model (ResNet50) trained based on the ϕ-Net dataset and evaluated on the ϕ-Net dataset, (2) a second model (ResNet50) trained based on the Residential dataset (trained in the same way as the first model) and evaluated on the Residential dataset, and (3) the first model evaluated on the Residential dataset, with respect to the above-mentioned example task are shown in FIG. 16. From FIG. 16, it can be seen that both the first and second models are able to obtain satisfactory performances when applied on their respective training datasets (i.e., applied on the same training dataset which they were trained on). However, when the first model was applied to the Residential dataset, its performance deteriorated significantly. In particular, for the example task, the success rate of the first model on input images from the Residential dataset is below 70%. Accordingly, this experiment demonstrated the unsatisfactory performance on a trained model when applied in a new scenario/environment that is materially different from the training dataset on which the model was trained on, and thus, such a trained model may not perform as expected in practical scenarios (i.e., real-world scenarios). In this regard, to address such a generalization problem, various example embodiments seek to produce an environment-adaptive learning model that is able to produce satisfactory or good performance even when applied in a new scenario/environment.

[00103] FIG. 17 depicts a schematic drawing of the heterogeneous learning framework or model 1300 that is further subjected to environment adaptive learning according to various example embodiments of the present invention. In this regard, in various example embodiments, the heterogeneous learning further comprises performing fine-tuning of the network portions 'a' and 'b' 1310, 1320 comprising: performing, for each labelled image of a subset of labelled images of the training dataset, a low-frequency spectral alignment of the labelled image with respect to an image of a subset of images of a second dataset (e.g., a testing dataset associated with a new scenario/environment) to obtain a low-frequency aligned labelled image, thereby

obtaining a low-frequency aligned training data subset. Accordingly, in various example embodiments, the training dataset and the second dataset are obtained from different sources.

[00104] In various example embodiments, the above-mentioned performing, for each labelled image of the subset of labelled images of the training dataset, the low-frequency spectral alignment of the labelled image with respect to the image of the subset of images of the second dataset comprises: obtaining a Fourier transform of the labelled image and a Fourier transform of the image; obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the labelled image and obtaining a magnitude spectrum of the Fourier transform of the image; and performing the low-frequency spectral alignment of the labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image, the magnitude spectrum of Fourier transform of the image and a first alignment parameter.

[00105] In various example embodiments, the above-mentioned performing the low-frequency spectral alignment of the labelled image comprises: modifying the magnitude spectrum of the Fourier transform of the labelled image based on the magnitude spectrum of the Fourier transform of the image and the first alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the labelled image; and performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image to obtain the low-frequency aligned labelled image.

[00106] In various example embodiments, the first alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the image obtained for modifying the magnitude spectrum of the Fourier transform of the labelled image. In various example embodiments, the first alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

[00107] In various example embodiments, the above-mentioned performing fine-tuning of the network portions ‘a’ and ‘b’ 1310, 1320 further comprises: performing, for each image of a plurality of images of the subset of images of the second dataset, self-supervised labelling of the image to obtain a pseudo-labelled image, thereby obtaining a subset of pseudo-labelled images; and performing, for each pseudo-labelled image of the subset of pseudo-labelled images, a low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the training dataset to obtain a low-frequency aligned pseudo-labelled image, thereby obtaining a low-frequency aligned second data subset.

[00108] In various example embodiments, the above-mentioned performing, for each pseudo-labelled image of the subset of pseudo-labelled images, the low-frequency spectral alignment of the pseudo-labelled image with respect to the labelled image of the subset of labelled images of the training dataset comprises: obtaining a Fourier transform of the pseudo-labelled image and a Fourier transform of the labelled image; obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the pseudo-labelled image and obtaining a magnitude spectrum of the Fourier transform of the labelled image; and performing the low-frequency spectral alignment of the pseudo-labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image, the magnitude spectrum of Fourier transform of the labelled image and a second alignment parameter.

[00109] In various example embodiments, the above-mentioned performing the low-frequency spectral alignment of the pseudo-labelled image comprises: modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image based on the magnitude spectrum of the Fourier transform of the labelled image and the second alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the pseudo-labelled image; and performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image to obtain the low-frequency aligned pseudo-labelled image.

[00110] In various example embodiments, the second alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the labelled image obtained for modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image. In various example embodiments, the second alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

[00111] In various example embodiments, the above-mentioned performing fine-tuning of the network portions ‘a’ and ‘b’ 1310, 1320 comprises performing a plurality of fine-tuning rounds. Each of the plurality of fine-tuning rounds comprises: shuffling the labelled images in the training dataset and the images in the second dataset; extracting a plurality of subsets of images from the second dataset, and for each subset of images, extracting a subset of labelled images from the training dataset to form a subset pair of the subset of images and the subset of labelled images, thereby forming a plurality of subset pairs; and for each subset pair of the plurality of subset pairs: performing, for each labelled image of the subset of labelled images of the subset pair, the above-mentioned low-frequency spectral alignment of the labelled image

with respect to an image of the subset of images of the subset pair to obtain the low-frequency aligned labelled image, thereby obtaining the low-frequency aligned training data subset; performing, for each image of a plurality of images of the subset of images of the subset pair, the above-mentioned self-supervised labelling of the image to obtain the pseudo-labelled image, thereby obtaining the subset of pseudo-labelled images; performing, for each pseudo-labelled image of the subset of pseudo-labelled images, the above-mentioned low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the subset pair to obtain the low-frequency aligned pseudo-labelled image, thereby obtaining the low-frequency aligned second data subset; and training the network portions 'a' and 'b' 1310, 1320 based on the low-frequency aligned training data subset and the low-frequency aligned second data subset.

[00112] Accordingly, in various example embodiments, the network portions 'a' and 'b' 1310, 1320 are further fine-tuned based on low-frequency aligned training data subsets and low-frequency aligned second data subsets (e.g., low-frequency aligned testing data subsets), for improving/enhancing the usability of the model 1300 in image classification and image localization and/or segmentation when deployed in practical applications.

[00113] In various example embodiments, the network portion 'b' 1320 is configured to perform building defect localization and/or building defect segmentation, and the training dataset comprises labelled structural images for training the network portion 'a' 1310 to perform image classification and for training the network portion 'b' 1320 to perform building defect localization and/or building defect segmentation.

[00114] For illustration purpose only by way of an example, an example implementation of the environment adaptive learning technique will now be described according to various example embodiments of the present invention. In this regard, to address the above-mentioned generalization problem, an environment-adaptive learning technique is provided that dynamically addresses or mitigates model performance degradation in diverse environmental conditions. For example, various example embodiments found that primary variations in building defect images stem from external factors such as lighting and equipment, rather than differences in defect types. Furthermore, various example embodiments found that this distinction manifests in the low-frequency components of images originating from different sources. Therefore, various example embodiments advantageously seek to address or mitigate the above-mentioned generalization problem by aligning different datasets from different

sources/environments in low-frequency components to improve the performance of the model 1300 even when applied in a new scenario/environment.

[00115] In various example embodiments, to align the low-frequency components of images, Fourier transform (Discrete Fourier transform) is employed. In particular, for an image with dimensions of length M and width N, whereby each pixel of the image is denoted as $f(m,n)$, a two-dimensional Fourier Transform may be performed on the image to transform it as follows:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(m,n)e^{-2j\pi}(\frac{um}{M}+\frac{vn}{N})$$
$$= R(u,v) + I(u,v)j$$

(Equation 3)

[00116] The corresponding frequency spectrum (which may also be referred to as the magnitude spectrum) of Fourier transform (denoted as *A*) and the phase spectrum (denoted as *P*) may be obtained as follows:

$$A(u,v) = \sqrt{R(u,v)^2 + I(u,v)^2}$$

(Equation 4)

$$P(u,v) = arctan\,(\frac{I(u,v)}{R(u,v)})$$

(Equation 5)

[00117] Furthermore, the inverse Fourier transform may be defined as:

$$f(x,y) = F^{-1}(A(u,v), P(u,v))$$

(Equation 6)

[00118] To align the image in the low-frequency part, an alignment coefficient (denoted as *K*) may be introduced as follows:

$$K = \begin{cases} 0.95, (u,v) \in L \\ 0.05, (u,v) \notin L \end{cases}$$

(Equation 7)

where *L* refers to the low-frequency region of the image, which may refer to the lowest 10% of frequency components of the image. It will be appreciated by a person skilled in the art that the lowest frequency components of an image are not limited to the lowest 10% of frequency components and may be varied as desired or as appropriate. In various example embodiments, a frequency point is determined to correspond to a low-frequency point if the frequency point corresponds to a frequency component in the lowest 1% to the lowest 10% (i.e., $\beta \in [0.01, 0.1]$) of frequency components of an image being subjected to the low-frequency spectral alignment.

In this regard, it was found that values greater than 10% or less than 1% often lead to a decrease in performance. It will also be appreciated that the value of the alignment coefficient K is not limited to 0.95 for a low frequency point and 0.05 otherwise as shown in Equation (7), and may be varied as desired or as appropriate. In this regard, experiments were conducted to evaluate the model performance against the value of the alignment coefficient K (for a low frequency point), and the experimental results are shown in a plot in FIG. 18. In this regard, in various example embodiments, the value of the alignment coefficient K (for a low frequency point) may be in a range between 0.9 and 1, more preferably, between 0.93 to 0.98, and more preferably, between 0.94 to 0.97. Therefore, the value of the alignment coefficient K may be a hyperparameter and may be determined experimentally. In various example embodiments, the above-mentioned first and second alignment parameters may be the same alignment parameter (the same alignment coefficient K).

[00119] For illustration purpose and by way an example only, FIGs. 19A and 19B show a one-dimensional signal and a Fourier transform of the one-dimensional signal, respectively, according to various example embodiments of the present invention. For example, assuming that a frequency point is determined to correspond to a low-frequency point if the frequency point corresponds to a frequency component in the lowest 10% (i.e., $\beta = 0.1$) of frequency components of an image being subjected to the low-frequency spectral alignment, from the frequency components shown in FIG. 19B, any frequency point on or below 50 Hz (lowest 10% of the frequency components up to 500 Hz) is considered to be a low-frequency point. For example, one-dimensional signals and two-dimensional signals may both be stored as matrices in a computer. A one-dimensional signal may be represented as a one-dimensional matrix, while a two-dimensional image may either be represented as a two-dimensional matrix (e.g., in the case of grayscale images) or a three-dimensional matrix (e.g., in the case of color images). Therefore, in the case of one-dimensional matrices, one-dimensional Fourier transforms may be employed, while for two-dimensional matrices, two-dimensional Fourier transforms may be employed. Accordingly, it can be determined whether a frequency point being processed corresponds to a low-frequency point in a low-frequency region of an image based on a matrix of the image stored in a computer.

[00120] In the example implementation of the environment adaptive learning technique according to various example embodiments, firstly, the training dataset may be aligned to the testing dataset in the low frequency region. For example, the ϕ-Net dataset may be the training dataset which was used to train the model 1300 prior to this environment adaptive learning

process and it may be intended to apply the model 1300 to the testing dataset (e.g., the above-mentioned Residential dataset). For any two images, one from the training dataset and the other from the testing dataset, denoted as x^{tr} and x^{te} respectively, aligning the low-frequency components of the image x^{tr} to the image x^{te} may be performed as follows:

$$x^{tr\to te} = F^{-1}((K * A^{te} + (1-K)A^{tr}), P^{tr})$$

(Equation 8)

[00121] Therefore, to perform Equation (8), a Fourier transform of the labelled image x^{tr} and a Fourier transform of the image x^{te} may be obtained, a magnitude spectrum A^{tr} and a phase spectrum P^{tr} of the Fourier transform of the labelled image x^{tr} may be obtained, a magnitude spectrum A^{te} of the Fourier transform of the image x^{te} may be obtained. The low-frequency spectral alignment of the labelled image x^{tr} may then be performed based on the magnitude spectrum A^{tr} and the phase spectrum P^{tr} of the Fourier transform of the labelled image x^{tr}, the magnitude spectrum A^{te} of Fourier transform of the image x^{te} and the alignment parameter K. In particular, as can be seen from Equation (8), the magnitude spectrum A^{tr} of the Fourier transform of the labelled image x^{tr} may be modified based on the magnitude spectrum A^{te} of the Fourier transform of the image x^{te} and the alignment parameter K to obtain a modified magnitude spectrum of the Fourier transform of the labelled image x^{tr}, and an inverse Fourier transform may then be performed based on the modified magnitude spectrum and the phase spectrum P^{tr} of the Fourier transform of the labelled image x^{tr} to obtain the low-frequency aligned labelled image $x^{tr\to te}$. Therefore, it can be understood that the alignment parameter K is configured to control an amount of the magnitude spectrum A^{te} of the Fourier transform of the image x^{te} obtained for modifying the magnitude spectrum A^{tr} of the Fourier transform of the labelled imagex^{tr}.

[00122] Accordingly, the transformed image $x^{tr\to te}$ is the result of the low-frequency spectral alignment, and the image labels on the transformed image remain unchanged. Therefore, the original image labels can continue to be used for training, that is, the associated loss function may be expressed as follows:

$$L_{ce} = L(x^{tr\to te}, y^{tr})$$

(Equation 9)

[00123] For illustration purpose and by way of an example only, FIGs. 20A and 20B show an example image before and after the Fourier Transform, according to various example embodiments of the present invention.

[00124] Accordingly, by performing the above operations, images x^{tr} from the training dataset can be aligned to images x^{te} from the testing dataset in the low frequency region according to various example embodiments of the present invention.

[00125] In various example embodiments, when performing training, a batch of images may be inputted to the model at a time (e.g., a batch of 64 images at a time, including a subset of 32 images from the training dataset and a subset of 32 images from the testing dataset). In this regard, the subset of 32 images from the training dataset and the subset of 32 images from the testing dataset may be sequentially paired to form a batch of 32 pairs of images. Equation (8) above may then be applied to this batch of 32 pairs of images. For example, assuming that the quantity of images in the training dataset is significantly larger than that of the test dataset (e.g., the testing dataset has only 320 test images). After performing the above operation (e.g., Equation (8)) 10 times on 10 batches of images, respectively, the testing dataset will be exhausted, thereby completing one training epoch. In various example embodiments, multiple training epochs (e.g., corresponding to the above-mentioned plurality of fine-tuning rounds, which may be many training epochs, such as 50 or more) are conducted, and at each training epoch (e.g., before or at a start thereof), the training dataset and the testing dataset are randomly shuffled and the above operations are repeated for the new training epoch.

[00126] Therefore, with a sufficient number of training epochs, all images (or substantively all images) in the training dataset can be covered despite the quantity of images in the training dataset being significantly larger than that of the testing dataset. In various example embodiments, if the distribution of the training dataset remains the same and is from the same source, it may not be necessary to fully traverse all images in the training dataset.

[00127] In various example embodiments, the testing dataset (e.g., the above-mentioned Residential dataset) may also be aligned to the training dataset (e.g., the original ϕ-Net dataset). In this regard, various example embodiments note that the testing dataset lacks labels, thereby rendering supervised learning for training impractical. To address this problem, various example embodiments apply self-supervised constraints to the model 1300.

[00128] For a given image x^{te}, when the maximum probability of the network's output (or the maximum probabilities of multiple outputs for multiple attributes, respectively) for the image x^{te} exceeds a predefined threshold, various example embodiments use this output as a pseudo-label $\hat{y}^{te}$, which may be expressed as follows:

$$\hat{y}^{te} = \theta_{tr}(x^{te})$$

(Equation 10)

where $\theta_{tr}(x^{te})$ denotes the model 1300 trained on the training dataset prior to this environment adaptive learning process. Subsequently, image x^{te} is aligned towards image x^{tr} in the low frequency region to bridge the alignment between the testing and training datasets, which may be performed as follows:

$$x^{te\to tr} = F^{-1}((K * A^{tr} + (1-K)A^{te}), P^{te})$$

(Equation 11)

[00129] Therefore, to perform Equation (11), a Fourier transform of the pseudo-labelled image x^{te} and a Fourier transform of the labelled image x^{tr} may be obtained, a magnitude spectrum A^{te} and a phase spectrum P^{te} of the Fourier transform of the pseudo-labelled image x^{te} may be obtained, and a magnitude spectrum A^{tr} of the Fourier transform of the labelled image x^{tr} may be obtained. The low-frequency spectral alignment of the pseudo-labelled image x^{te} may then be performed based on the magnitude spectrum A^{te} and the phase spectrum P^{te} of the Fourier transform of the pseudo-labelled image x^{te}, the magnitude spectrum A^{tr} of Fourier transform of the labelled image x^{tr} and the alignment parameter K. In particular, as can be seen from Equation (11), the magnitude spectrum A^{te} of the Fourier transform of the pseudo-labelled image x^{te} may be modified based on the magnitude spectrum A^{tr} of the Fourier transform of the labelled image x^{tr} and the alignment parameter K to obtain a modified magnitude spectrum of the Fourier transform of the pseudo-labelled image x^{te}, and an inverse Fourier transform may then be performed based on the modified magnitude spectrum and the phase spectrum P^{te} of the Fourier transform of the pseudo-labelled image x^{te} to obtain the low-frequency aligned pseudo-labelled image $x^{te\to tr}$. Therefore, it can be understood that the alignment parameter K is configured to also control an amount of the magnitude spectrum A^{tr} of the Fourier transform of the labelled image x^{tr} obtained for modifying the magnitude spectrum A^{te} of the Fourier transform of the pseudo-labelled image x^{te}.

[00130] In various example embodiments, the associated loss function may be expressed as follows:

$$L_{self} = \begin{cases} H(\hat{y}^{te}, \theta_{tr}(x_{te\to tr})) \\ 0 \end{cases}$$

(Equation 12)

where H denotes the cross entropy loss of pseudo-label $\hat{y}^{te}$ and the prediction of $\theta_{tr}(x_{te\to tr})$. If this loss is reduced, it implies that the variations in the low-frequency components have reduced impact on future predictions. The significance of this constraint is that it allows the low-frequency component of the testing dataset and the environment factors to become less

sensitive and impactful on future predictions by the model 1300 fine-tuned by this environment adaptive learning technique, and thus helps to mitigate overfitting.

[00131] For illustration purpose and by way of an example only, assuming that a test image has two attributes for a classification task, e.g., a first attribute being the spalling condition (spalling or non-spalling) and a second attribute being the component type (column, ceiling or wall), and the predefined threshold is set as 0.95. In this regard, if the predictions of these two attributes are [0.99, 0.01] (i.e., probabilities of spalling and non-spalling are 0.99 and 0.01, respectively) and [0.98, 0.01, 0.01] (i.e., probabilities of column, ceiling and wall are 0.98, 0.01, 0.01, respectively), the maximum probabilities of these two outputs for these two attributes are 0.99 and 0.98, respectively, which both satisfy the predefined threshold. Therefore, the pseudo-labels generated for the test image are 'spalling' and 'column' (i.e., the classes corresponding to the elements in the probability vectors with the maximum probabilities and satisfying (e.g., equal to or greater than) the predefine threshold), and the test image with the pseudo-labels is processed further according to Equations (11) and (12). On the other hand, for example, if the prediction of the first attribute is [0.5, 0.5], the maximum probability of the output for the first attribute does not satisfy the predefined threshold and thus will not be included or processed further (e.g., the test image will be removed or omitted).

[00132] As another example, for a segmentation task, the segmentation task may be conceptualized as a specialized form of classification where the task is to classify individual pixels. Each pixel is assigned to either the background or foreground, which, for example, may correspond to whether it represents a defect or not. For example, for each pixel in a test image, the prediction for the pixel may be represented as a probability vector. If there are only two classes, namely, background and foreground, the length of the probability vector is 2 and may be denoted as [p_foreground, p_background]. If the maximum probability associated with a class/element within the vector exceeds a predefined threshold, the pixel associated with the probability vector may then be assigned with a pseudo-label of the corresponding class.

[00133] The predefined threshold may be set as desired or as appropriate, and in various example embodiments, the predefined threshold is a value in a range from 0.9 to 0.99. It will be appreciated by a person skilled in the art that the present invention is not limited to the example techniques above for generating the pseudo-labels for the test images, and the pseudo-labels may be generated using any technique as desired or as appropriate.

[00134] Accordingly, by combining the loss function L_{ce} associated with aligning the training dataset to the testing dataset and the loss function L_{self} associated with aligning the testing dataset to the training dataset, the final loss function may be expressed as follow:

$$L_{final} = L_{ce} + L_{self}$$

(Equation 13)

[00135] Similarly, as explained above, when performing training, a batch of images may be inputted to the model at a time (e.g., a batch of 64 images at a time, including a subset of 32 images from the training dataset and a subset of 32 images from the testing dataset). For example, if the testing dataset has 330 test images, after shuffling randomly and drawing 32 images without replacement for each batch of images, a total of 320 test images will be used in 10 draws and leaving 10 images unused in this training epoch. As explained above, multiple training epochs (e.g., corresponding to the above-mentioned plurality of fine-tuning rounds, which may be many training epochs) are conducted, and at each training epoch (e.g., at a start thereof), the training dataset and the testing dataset are randomly shuffled, and the process is repeated by performing 10 more draws of 32 images each from the testing dataset. After multiple training epochs, because the testing dataset is shuffled before each training epoch, every single test data point is processed. In this regard, only the test images that can be used to generate pseudo-labels contribute to the model's optimization.

[00136] Therefore, the above-described environment-adaptive optimization objective advantageously takes into account the following two aspects:

- aligning the training dataset towards the testing dataset, thereby ensuring that the training dataset (e.g., the ϕ-Net dataset) aligns with the testing dataset (e.g., the Residential dataset) with respect to various factors such as lighting conditions, etc. This helps the model adapt to the characteristics of the testing dataset, which may be from a different source (e.g., environment).
- aligning the testing dataset towards the training dataset, thereby ensuring that the testing dataset is considered in the context of the training dataset, which advantageously makes the model 1300 less sensitive to different environmental factors.

[00137] For example, testing dataset often comes from a different distribution/environment than the training dataset. In this regard, various example embodiments seek to produce a trained model that performs well when deployed on the testing dataset as well. Accordingly, various example embodiments advantageously optimize the model 1300 without requiring original labels for the testing dataset. By way of an example practical application and without limitation,

assuming a model is trained using a training dataset from Singapore and the trained model is then deployed in another country such as in the United Kingdom with a significantly different environment. In this regard, the above-described environment-adaptive optimization technique advantageously enables the collection of some testing data in the new environment without requiring annotation/labeling (thus being simple and cost-effective) for optimizing the model for the new environment. For example, initially, a small amount of test data may be collected from the new environment for environment adaptive learning, and as the model is deployed and in use, new test data may be collected periodically (e.g., daily) and thus the model may be optimized periodically. Accordingly, over time, a large amount of unlabeled testing data may be collected and used to optimize the model, in an automated and continuous manner. In other words, the model continues to learn, optimize and improve its performance in the new environment over time.

[00138] Accordingly, the above-described environment-adaptive optimization objective seeks to balance the adaptation of the model 1300 both towards the training dataset and the testing dataset, creating a robust performance across different environmental factors. Therefore, in the heterogeneous learning according to various example embodiments of the present invention, the model 1300 may firstly be trained on the training data according to the backbone sharing technique as described hereinbefore, and such a trained model 1300 (e.g., after convergence, which may be referred to herein as model θ_{tr}) may then be fine-tuned by L_{final} according to the environment adaptive learning technique as described hereinbefore to obtain the model θ_{te}. In this regard, from experiments conducted, the model θ_{te} demonstrated superior generalization on the testing data from new scenarios/environments. In various example embodiments, in the environment adaptive learning technique, the fine-tuning of the trained model 1300 using the low-frequency aligned training dataset (or data subset) and the low-frequency aligned testing dataset (or data subset) may be performed in the same or similar manner as the training of the model 1300 using the training dataset according to the backbone sharing technique as described hereinbefore according to various example embodiments of the present invention.

[00139] To demonstrate the effectiveness of the environment adaptive learning technique described above according to various example embodiments of the present invention, experiments were conducted using the environment adaptive optimized model (optimized using the above environment adaptive learning technique) compared with the original model trained only on the training data (i.e., the trained model prior to being optimized by the above

environment adaptive learning technique), on the unlabelled residential dataset. The experimental results are presented in Table 6, Table 7 and Table 8 shown in FIGs. 21A, 21B and 21C, respectively. In particular, Table 6 compares the classification performances (accuracy %) of the above-mentioned models, Table 7 compares the localization performances (accuracy %) of the above-mentioned models on the residential dataset and Table 8 compares the segmentation performances (accuracy %) of the above-mentioned models on the residential dataset.

[00140] Therefore, it has been demonstrated that directly applying a trained model that has been trained only on a training dataset on a testing dataset (e.g., obtained from a different source/environment) may result in a significant drop in performances of the trained model. Furthermore, it has been demonstrated that the environment adaptive learning/optimization method according to various example embodiments can significantly mitigate such a problem. In particular, as shown in FIGs. 21A to 21C, for classification, localization and segmentation tasks, the environment adaptive optimized models perform much better than deploying the trained model directly into a new environment without the environment adaptive learning/optimization. Therefore, the environment adaptive learning/optimization method advantageously enhances the model's generalization capabilities, thereby increasing its practical applicability.

[00141] Accordingly, various example embodiments provide a method of training a neural network method comprises two stages. At a first stage, a weight sharing technique as described hereinbefore may be employed. For example, referring to the example architecture 1400 shown in FIG. 14, for any given training image, it is fed into two identical Swin Transformer networks 1412, 1422 for feature extraction. The extracted features may then be used for both classification and localization/segmentation tasks. Through backpropagation and weight sharing, the weight parameters of the two Swin Transformers 1412, 1422 are updated. The Swin Transformers 1412, 1422 each takes an image as input and produces features as output. The multi-task classifiers 1414 may each receives extracted deep features from the Swin Transformer 1412 as input and produces a classification result as output, while the mask predictor 1426 receives extracted deep features from the Swin Transformer 1422 as input and produces a segmentation result as output. At a second stage, an environment adaptive learning/optimization technique as described hereinbefore may be employed. As described hereinbefore, Fourier transforms may be utilized to adapt the model to a testing dataset. In particular, the environment adaptive learning/optimization technique includes aligning the

training dataset with the testing dataset in the low-frequency domain, as well as aligning the testing dataset with the training dataset in the low-frequency domain. Therefore, different datasets from different sources are aligned in the low-frequency domain. Such a technique advantageously mitigates the generalization issues caused by environmental factors such as lighting and architectural styles, which could otherwise lead to a significant decline in model performance. As a result, such a technique significantly enhanced performance in real-world applications. Accordingly, in various example embodiments, an integrated/unified model for performing image classification and image localization and/or segmentation is provided, along with various advantages as described hereinbefore.

[00142] While embodiments of the invention have been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A method of training a neural network model using at least one processor, the neural network model comprising a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or segmentation, the method comprising:

training, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round;

applying, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion;

training, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round; and

performing, after the first training round, a plurality of additional training rounds, each additional training round comprising:

updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round;

training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion corresponding to the additional training round;

updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round; and

training, at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round.

2. The method according to claim 1, wherein

said updating, at the additional training round, the first backbone parameters of the first neural network portion is further based on a first backbone sharing parameter; and

said updating, at the additional training round, the second backbone parameters of the second neural network portion is further based on a second backbone sharing parameter.

3. The method according to claim 2, wherein

the first backbone sharing parameter is configured to control an amount of the second backbone parameters obtained for updating the first backbone parameters of the first neural network portion, and

the second backbone sharing parameter is configured to control an amount of the first backbone parameters obtained for updating the second backbone parameters of the second neural network portion.

4. The method according to any one of claims 1 to 3, wherein said performing the plurality of additional training rounds comprises performing a number of additional training rounds until each of the first and second neural network portions converges.

5. The method according to any one of claims 1 to 4, wherein

the first and second neural network portions each comprises a Swin Transformer as a backbone thereof,

the first backbone parameters comprise first weight parameters; and

the second backbone parameters comprise second weight parameters.

6. The method according to any one of claims 1 to 5, wherein

the first neural network portion comprises a plurality of task classifiers configured to perform a plurality of classification tasks, respectively,

the training dataset comprises labelled images comprising multi-attribute labelled images, and

the first neural network portion is trained for the plurality of classification tasks simultaneously.

7. The method according to any one of claims 1 to 6, further comprising performing fine-tuning of the first and second neural network portions comprising:

performing, for each labelled image of a subset of labelled images of the training dataset, a low-frequency spectral alignment of the labelled image with respect to an image of a subset

of images of a second dataset to obtain a low-frequency aligned labelled image, thereby obtaining a low-frequency aligned training data subset.

8. The method according to claim 7, wherein said performing, for each labelled image of the subset of labelled images of the training dataset, the low-frequency spectral alignment of the labelled image with respect to the image of the subset of images of the second dataset comprises:

obtaining a Fourier transform of the labelled image and a Fourier transform of the image;

obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the labelled image and obtaining a magnitude spectrum of the Fourier transform of the image; and

performing the low-frequency spectral alignment of the labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image, the magnitude spectrum of Fourier transform of the image and a first alignment parameter.

9. The method according to claim 8, wherein said performing the low-frequency spectral alignment of the labelled image comprises:

modifying the magnitude spectrum of the Fourier transform of the labelled image based on the magnitude spectrum of the Fourier transform of the image and the first alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the labelled image; and

performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image to obtain the low-frequency aligned labelled image.

10. The method according to claim 8 or 9, wherein

the first alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the image obtained for modifying the magnitude spectrum of the Fourier transform of the labelled image, and

the first alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

11. The method according to any one of claims 7 to 10, wherein said performing fine-tuning of the first and second neural network portions further comprises:

performing, for each image of a plurality of images of the subset of images of the second dataset, self-supervised labelling of the image to obtain a pseudo-labelled image, thereby obtaining a subset of pseudo-labelled images; and

performing, for each pseudo-labelled image of the subset of pseudo-labelled images, a low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the training dataset to obtain a low-frequency aligned pseudo-labelled image, thereby obtaining a low-frequency aligned second data subset.

12. The method according to claim 11, wherein said performing, for each pseudo-labelled image of the subset of pseudo-labelled images, the low-frequency spectral alignment of the pseudo-labelled image with respect to the labelled image of the subset of labelled images of the training dataset comprises:

obtaining a Fourier transform of the pseudo-labelled image and a Fourier transform of the labelled image;

obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the pseudo-labelled image and obtaining a magnitude spectrum of the Fourier transform of the labelled image; and

performing the low-frequency spectral alignment of the pseudo-labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image, the magnitude spectrum of Fourier transform of the labelled image and a second alignment parameter.

13. The method according to claim 12, wherein said performing the low-frequency spectral alignment of the pseudo-labelled image comprises:

modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image based on the magnitude spectrum of the Fourier transform of the labelled image and the second alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the pseudo-labelled image; and

performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image to obtain the low-frequency aligned pseudo-labelled image.

14. The method according to claim 12 or 13, wherein

the second alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the labelled image obtained for modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image, and

the second alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

15. The method according to any one of claims 11 to 14, wherein said performing fine-tuning of the first and second neural network portions comprises performing a plurality of fine-tuning rounds, each of the plurality of fine-tuning rounds comprising:

shuffling the labelled images in the training dataset and the images in the second dataset;

extracting a plurality of subsets of images from the second dataset, and for each subset of images, extracting a subset of labelled images from the training dataset to form a subset pair of the subset of images and the subset of labelled images, thereby forming a plurality of subset pairs; and

for each subset pair of the plurality of subset pairs:

performing, for each labelled image of the subset of labelled images of the subset pair, said low-frequency spectral alignment of the labelled image with respect to an image of the subset of images of the subset pair to obtain the low-frequency aligned labelled image, thereby obtaining the low-frequency aligned training data subset;

performing, for each image of a plurality of images of the subset of images of the subset pair, said self-supervised labelling of the image to obtain the pseudo-labelled image, thereby obtaining the subset of pseudo-labelled images;

performing, for each pseudo-labelled image of the subset of pseudo-labelled images, said low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the subset pair to obtain the low-frequency aligned pseudo-labelled image, thereby obtaining the low-frequency aligned second data subset; and

training the first and second neural network portions based on the low-frequency aligned training data subset and the low-frequency aligned second data subset.

16. The method according to claim 10 or 14, wherein the frequency point is determined to correspond to a low-frequency point if the frequency point corresponds to a frequency component in a lowest 1% to a lowest 10% of frequency components of an image being subjected to the low-frequency spectral alignment.

17. The method according to any one of claims 7 to 16, wherein the training dataset and the second dataset are obtained from different sources.

18. The method according to any one of claims 1 to 17, wherein

the second neural network portion is configured to perform building defect localization and/or building defect segmentation, and

the training dataset comprises labelled structural images for training the first neural network portion to perform image classification and for training the second neural network portion to perform building defect localization and/or building defect segmentation.

19. A system for training a neural network model, the neural network model comprising a first neural network portion configured to perform image classification and a second neural network portion configured to perform image localization and/or segmentation, the system comprising:

at least one memory; and

at least one processor communicatively coupled to the at least one memory and configured to:

train, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round;

apply, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion;

train, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round; and

perform, after the first training round, a plurality of additional training rounds, each additional training round comprising:

updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous additional training round;

training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion corresponding to the additional training round;

updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round; and

training, at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round.

20. The system according to claim 19, wherein

said updating, at the additional training round, the first backbone parameters of the first neural network portion is further based on a first backbone sharing parameter; and

said updating, at the additional training round, the second backbone parameters of the second neural network portion is further based on a second backbone sharing parameter.

21. The system according to claim 20, wherein

the first backbone sharing parameter is configured to control an amount of the second backbone parameters obtained for updating the first backbone parameters of the first neural network portion, and

the second backbone sharing parameter is configured to control an amount of the first backbone parameters obtained for updating the second backbone parameters of the second neural network portion.

22. The system according to any one of claims 19 to 21, wherein said perform the plurality of additional training rounds comprises performing a number of additional training rounds until each of the first and second neural network portions converges.

23. The system according to any one of claims 19 to 22, wherein

the first and second neural network portions each comprises a Swin Transformer as a backbone thereof,

the first backbone parameters comprise first weight parameters; and

the second backbone parameters comprise second weight parameters.

24. The system according to any one of claims 19 to 23, wherein

the first neural network portion comprises a plurality of task classifiers configured to perform a plurality of classification tasks, respectively,

the training dataset comprises labelled images comprising multi-attribute labelled images, and

the first neural network portion is trained for the plurality of classification tasks simultaneously.

25. The system according to any one of claims 19 to 24, wherein the at least one processor is further configured to perform fine-tuning of the first and second neural network portions comprising:

perform, for each labelled image of a subset of labelled images of the training dataset, a low-frequency spectral alignment of the labelled image with respect to an image of a subset of images of a second dataset to obtain a low-frequency aligned labelled image, thereby obtaining a low-frequency aligned training data subset.

26. The system according to claim 25, wherein said perform, for each labelled image of the subset of labelled images of the training dataset, the low-frequency spectral alignment of the labelled image with respect to the image of the subset of images of the second dataset comprises:

obtaining a Fourier transform of the labelled image and a Fourier transform of the image;

obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the labelled image and obtaining a magnitude spectrum of the Fourier transform of the image; and

performing the low-frequency spectral alignment of the labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image, the magnitude spectrum of Fourier transform of the image and a first alignment parameter.

27. The system according to claim 26, wherein said performing the low-frequency spectral alignment of the labelled image comprises:

modifying the magnitude spectrum of the Fourier transform of the labelled image based on the magnitude spectrum of the Fourier transform of the image and the first alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the labelled image; and

performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the labelled image to obtain the low-frequency aligned labelled image.

28. The system according to claim 26 or 27, wherein

the first alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the image obtained for modifying the magnitude spectrum of the Fourier transform of the labelled image, and

the first alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

29. The system according to any one of claims 25 to 28, wherein said perform fine-tuning of the first and second neural network portions further comprises:

perform, for each image of a plurality of images of the subset of images of the second dataset, self-supervised labelling of the image to obtain a pseudo-labelled image, thereby obtaining a subset of pseudo-labelled images; and

perform, for each pseudo-labelled image of the subset of pseudo-labelled images, a low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the training dataset to obtain a low-frequency aligned pseudo-labelled image, thereby obtaining a low-frequency aligned second data subset.

30. The system according to claim 29, wherein said perform, for each pseudo-labelled image of the subset of pseudo-labelled images, the low-frequency spectral alignment of the pseudo-labelled image with respect to the labelled image of the subset of labelled images of the training dataset comprises:

obtaining a Fourier transform of the pseudo-labelled image and a Fourier transform of the labelled image;

obtaining a magnitude spectrum and a phase spectrum of the Fourier transform of the pseudo-labelled image and obtaining a magnitude spectrum of the Fourier transform of the labelled image; and

performing the low-frequency spectral alignment of the pseudo-labelled image based on the magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image, the magnitude spectrum of Fourier transform of the labelled image and a second alignment parameter.

31. The system according to claim 30, wherein said performing the low-frequency spectral alignment of the pseudo-labelled image comprises:

modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image based on the magnitude spectrum of the Fourier transform of the labelled image and the second alignment parameter to obtain a modified magnitude spectrum of the Fourier transform of the pseudo-labelled image; and

performing an inverse Fourier transform based on the modified magnitude spectrum and the phase spectrum of the Fourier transform of the pseudo-labelled image to obtain the low-frequency aligned pseudo-labelled image.

32. The system according to claim 30 or 31, wherein

the second alignment parameter is configured to control an amount of the magnitude spectrum of the Fourier transform of the labelled image obtained for modifying the magnitude spectrum of the Fourier transform of the pseudo-labelled image, and

the second alignment parameter is defined as a function of a frequency point in a Fourier space and based on whether the frequency point corresponds to a low-frequency point.

33. The system according to any one of claims 29 to 32, wherein said perform fine-tuning of the first and second neural network portions comprises performing a plurality of fine-tuning rounds, each of the plurality of fine-tuning rounds comprising:

shuffling the labelled images in the training dataset and the images in the second dataset;

extracting a plurality of subsets of images from the second dataset, and for each subset of images, extracting a subset of labelled images from the training dataset to form a subset pair of the subset of images and the subset of labelled images, thereby forming a plurality of subset pairs; and

for each subset pair of the plurality of subset pairs:

performing, for each labelled image of the subset of labelled images of the subset pair, said low-frequency spectral alignment of the labelled image with respect to an image of the subset of images of the subset pair to obtain the low-frequency aligned labelled image, thereby obtaining the low-frequency aligned training data subset;

performing, for each image of a plurality of images of the subset of images of the subset pair, said self-supervised labelling of the image to obtain the pseudo-labelled image, thereby obtaining the subset of pseudo-labelled images;

performing, for each pseudo-labelled image of the subset of pseudo-labelled images, said low-frequency spectral alignment of the pseudo-labelled image with respect to a labelled image of the subset of labelled images of the subset pair to obtain the low-frequency aligned pseudo-labelled image, thereby obtaining the low-frequency aligned second data subset; and

training the first and second neural network portions based on the low-frequency aligned training data subset and the low-frequency aligned second data subset.

34. The system according to claim 28 or 32, wherein the frequency point is determined to correspond to a low-frequency point if the frequency point corresponds to a frequency component in a lowest 1% to a lowest 10% of frequency components of an image being subjected to the low-frequency spectral alignment.

35. The system according to any one of claims 19 to 34, wherein the training dataset and the second dataset are obtained from different sources.

36. The system according to any one of claims 19 to 35, wherein

the second neural network portion is configured to perform building defect localization and/or building defect segmentation, and

the training dataset comprises labelled structural images for training the first neural network portion to perform image classification and for training the second neural network portion to perform building defect localization and/or building defect segmentation.

37. A computer program product, embodied in one or more non-transitory computer-readable storage mediums, comprising instructions executable by at least one processor to perform the method of training a neural network model according to any one of claims 1 to 18.

38. A method of using the neural network model trained according to any one of claims 1 to 18, the method comprising:

receiving, by the neural network model, an input image;

performing, by the neural network model, image classification of the input image; and

performing, by the neural network model, image localization and/or segmentation of the input image,

wherein said performing image classification of the input image and said performing image localization and/or segmentation of the input image are performed simultaneously.

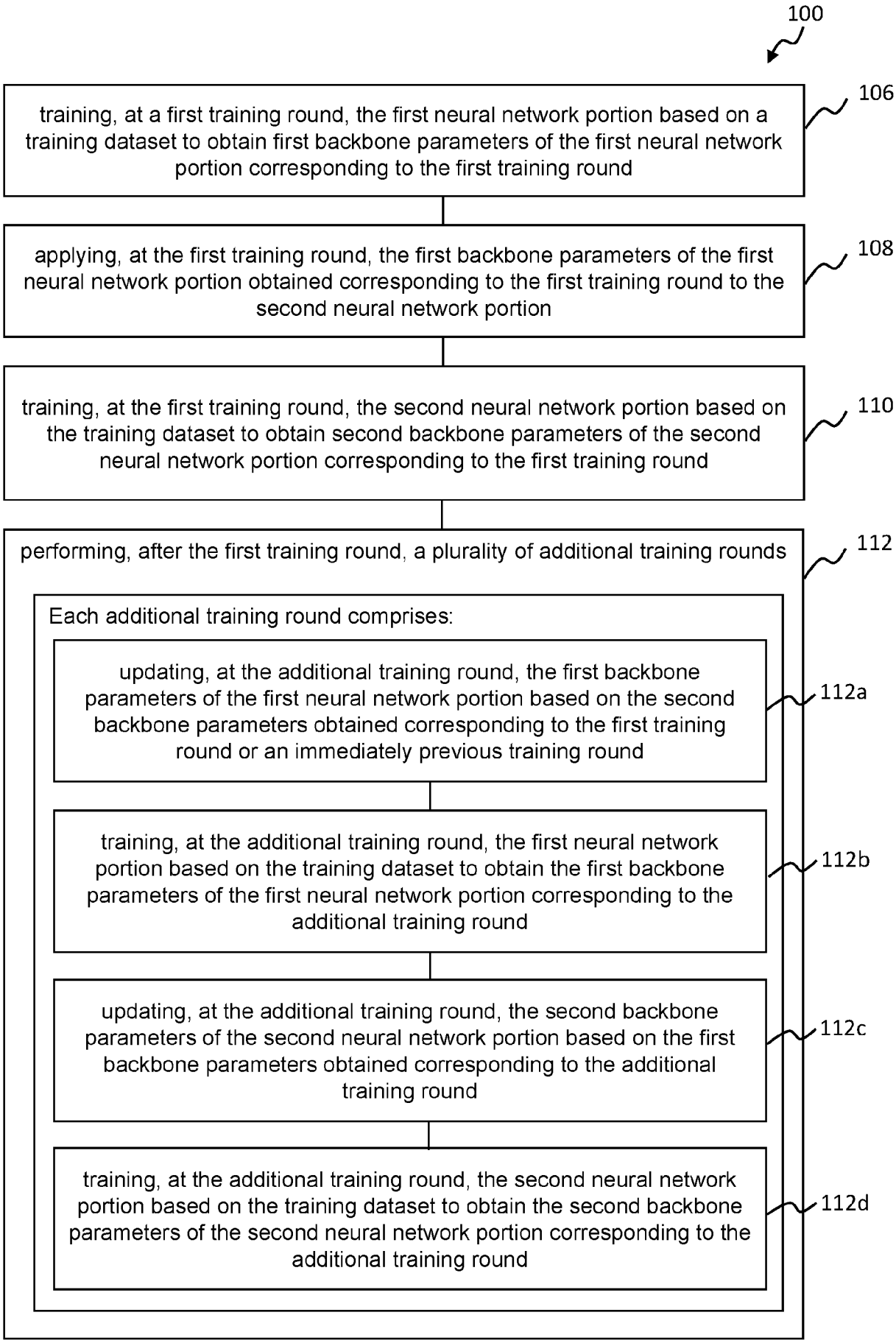

100
training, at a first training round, the first neural network portion based on a training dataset to obtain first backbone parameters of the first neural network portion corresponding to the first training round
106
applying, at the first training round, the first backbone parameters of the first neural network portion obtained corresponding to the first training round to the second neural network portion
108
training, at the first training round, the second neural network portion based on the training dataset to obtain second backbone parameters of the second neural network portion corresponding to the first training round
110
performing, after the first training round, a plurality of additional training rounds
112
Each additional training round comprises:
updating, at the additional training round, the first backbone parameters of the first neural network portion based on the second backbone parameters obtained corresponding to the first training round or an immediately previous training round
112a
training, at the additional training round, the first neural network portion based on the training dataset to obtain the first backbone parameters of the first neural network portion corresponding to the additional training round
112b
updating, at the additional training round, the second backbone parameters of the second neural network portion based on the first backbone parameters obtained corresponding to the additional training round
112c
training, at the additional training round, the second neural network portion based on the training dataset to obtain the second backbone parameters of the second neural network portion corresponding to the additional training round
112d

FIG. 1

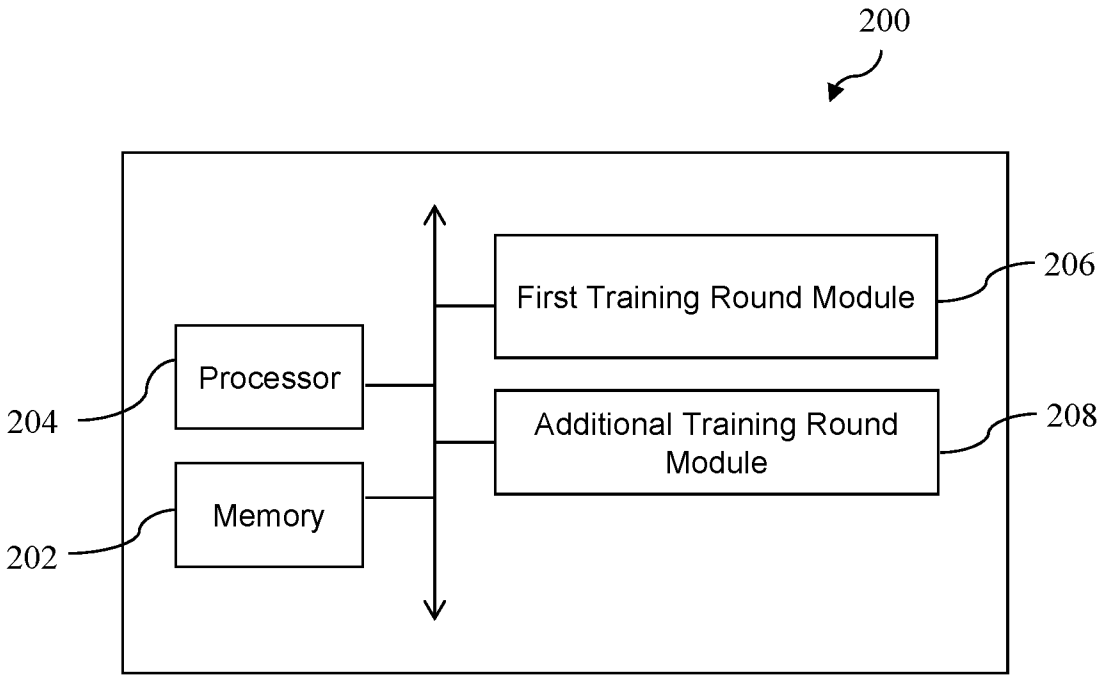

200
First Training Round Module
206
Processor
204
Additional Training Round Module
208
Memory
202

FIG. 2

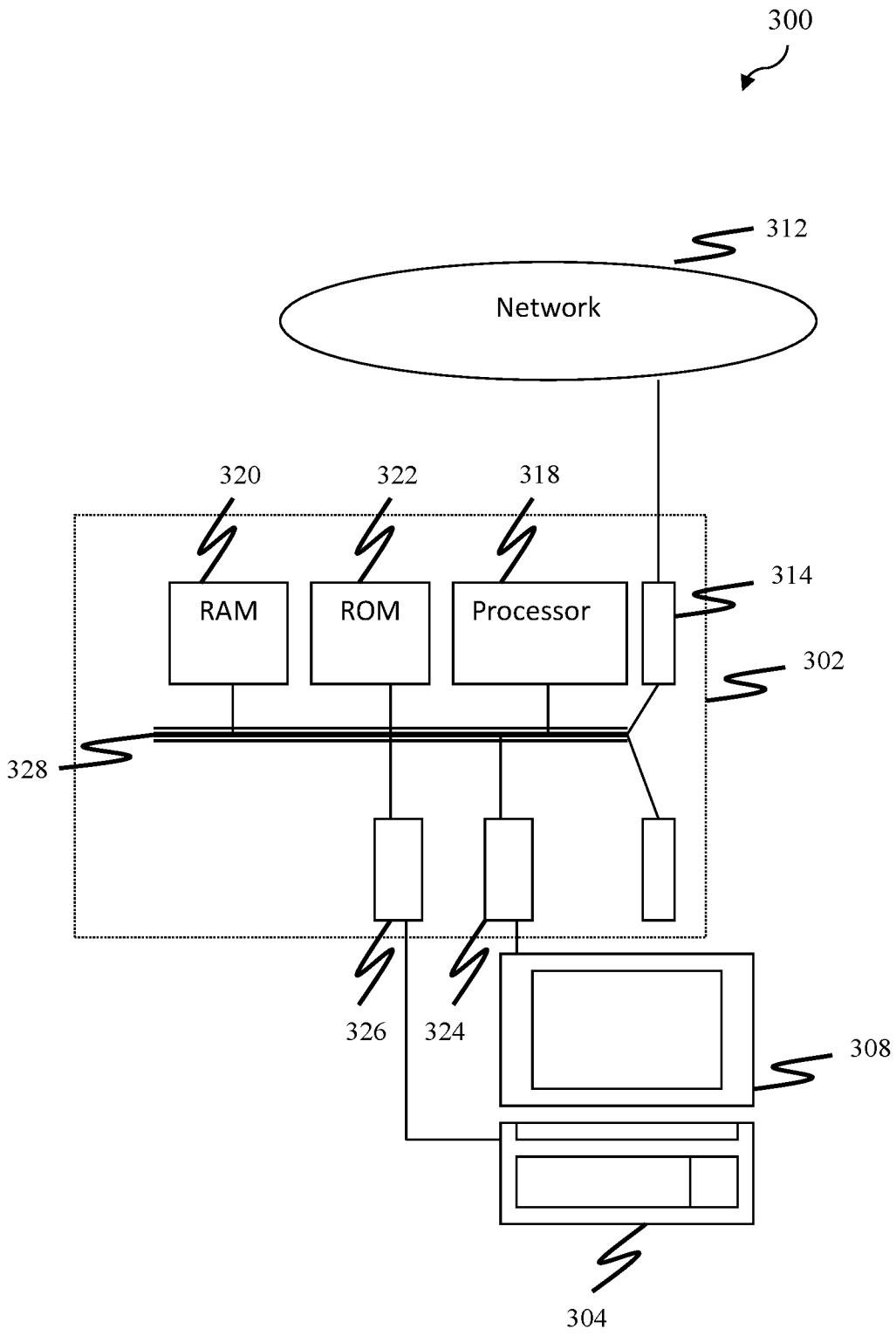

300
312
Network
320
322
318
RAM
ROM
Processor
314
302
328
326
324
308
304

FIG. 3

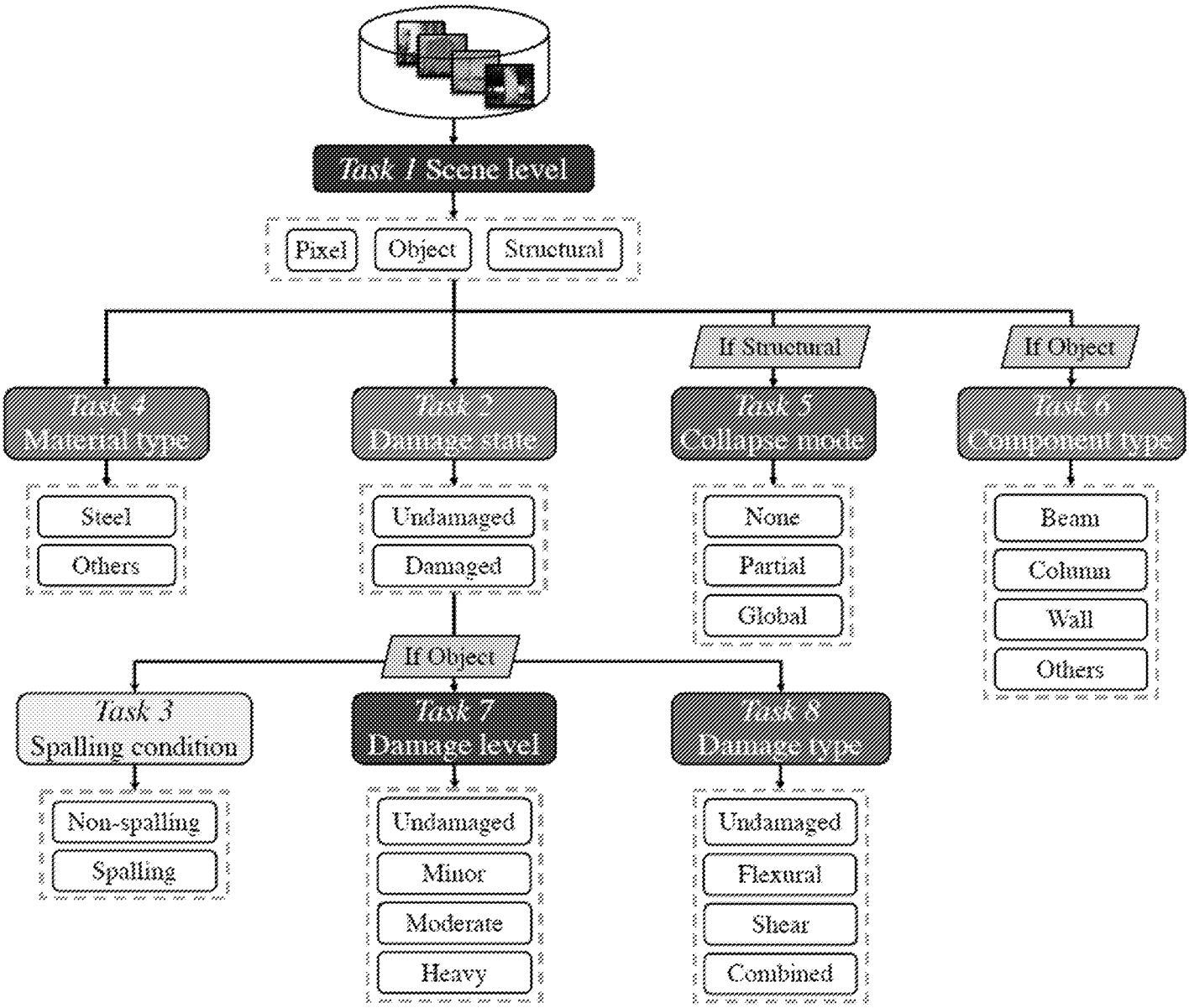

Task 1 Scene level
Pixel
Object
Structural
If Structural
If Object
Task 4
Material type
Task 2
Damage state
Task 5
Collapse mode
Task 6
Component type
Steel
Others
Undamaged
Damaged
None
Partial
Global
Beam
Column
Wall
Others
If Object
Task 3
Spalling condition
Task 7
Damage level
Task 8
Damage type
Non-spalling
Spalling
Undamaged
Minor
Moderate
Heavy
Undamaged
Flexural
Shear
Combined

FIG. 4

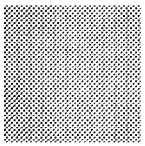
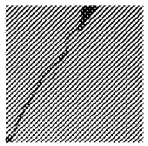
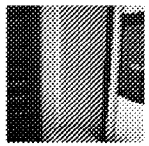
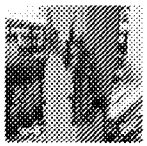
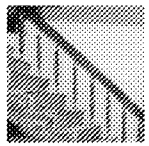
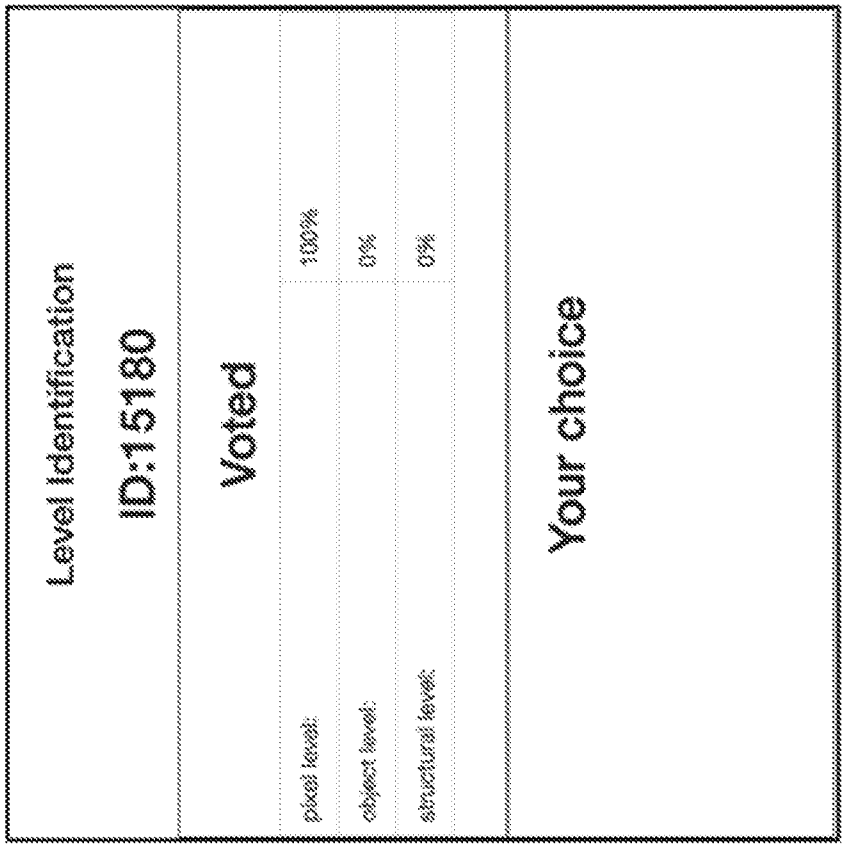
Task#										
1	Pixel	Pixel	Object	Object	Object	Object	Object	Structural	Structural	Structural
2	Undamaged	Damaged	Damaged	Damaged	Damaged	Damaged	Undamaged	Undamaged	Damaged	Damaged
3	Non-spalling	Non-spalling	Spalling	Non-spalling	Spalling	Non-spalling	Non-spalling	-	-	-
4	Others	Others	Others	Others	Others	Others	Others	Steel	Others	Others
5	-	-	-	-	-	-	-	Non-collapse	Partial-collapse	Global-collapse
6	-	-	Wall	Column	Column	Beam	Others	-		-
7	-	-	Combined	Flexural	Shear	Shear	-	-		-
8	-	-	Heavy	Minor	Heavy	Moderate	-	-		-

FIG. 5


Level Identification
ID:15180
Voted
pixel level:
100%
object level:
0%
structural level:
0%
Your choice

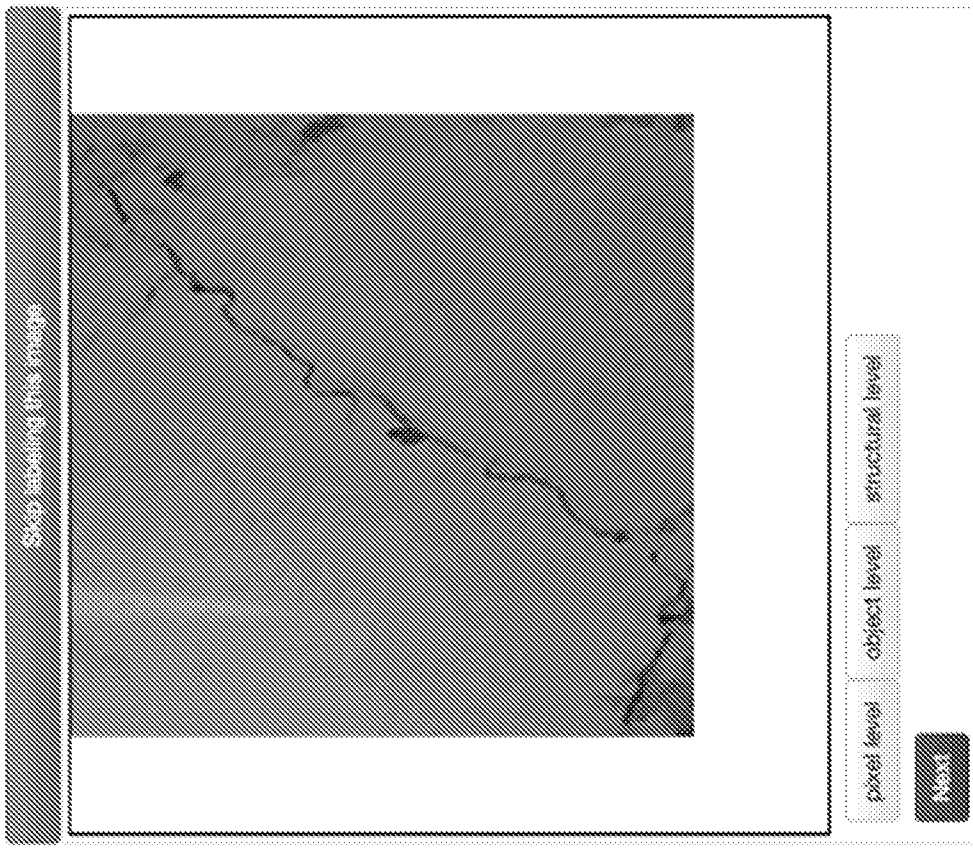

pixel level
object level
structural level
NEXT

FIG. 6

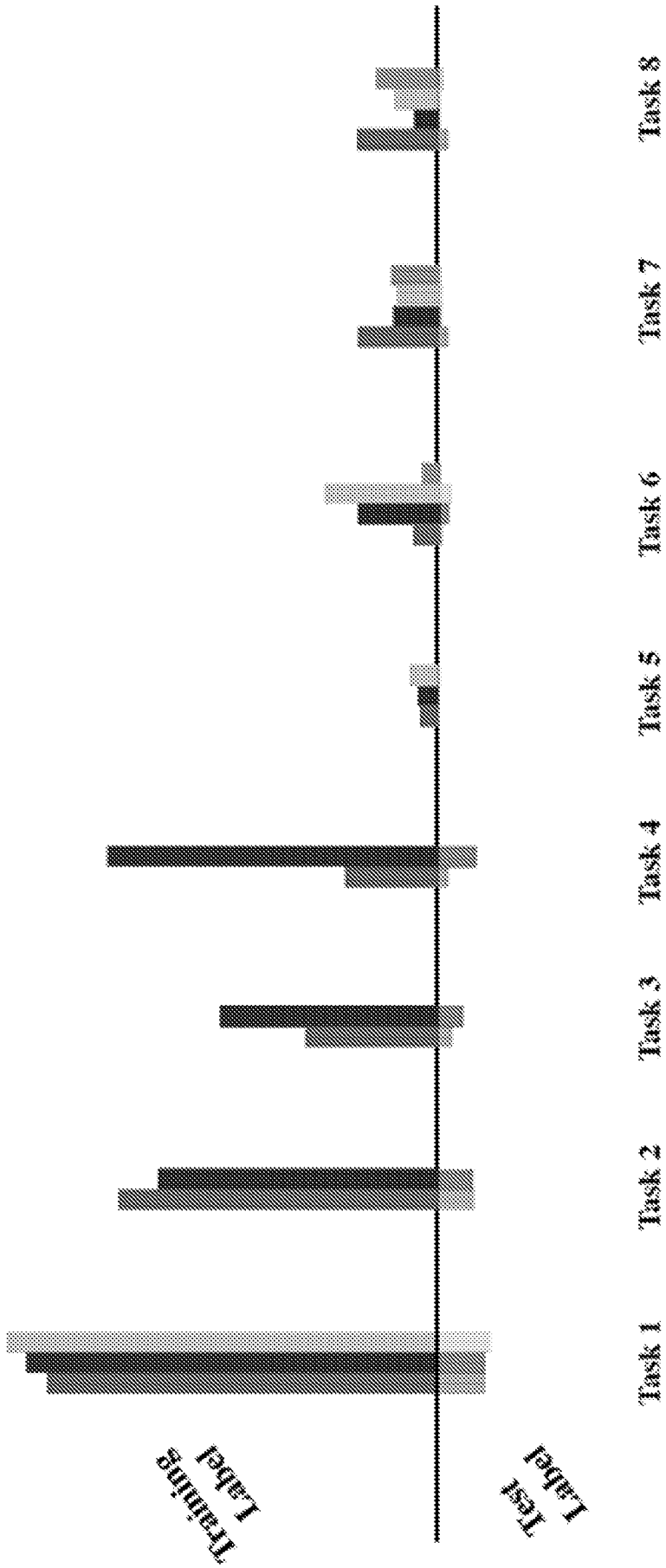

Training Label
Test Label
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8

FIG. 7

Table 1: Baseline results (%)

Model	VGG16		VGG19		ResNet50	
Task	BSL	BSL-DA	BSL	BSL-DA	BSL	BSL-DA
1	87.3	85.4	87.4	84.0	87.7	**88.0**
2	64.0	60.7	62.7	60.7	**76.7**	75.3
3	63.0	63.0	63.0	63.0	75.7	**77.1**
4	93.9	90.8	94.4	90.3	**95.1**	93.1
5	46.9	45.9	45.9	45.9	**65.1**	61.7
6	64.8	64.8	65.1	65.1	**69.8**	68.2
7	41.6	41.6	41.6	41.6	54.6	**58.4**
8	43.7	43.7	43.7	43.7	54.3	**56.1**

FIG. 8A

Table 2: Transfer learning results (%)

Model	VGG16		VGG19		ResNet50	
Task	TL	TL-DA	TL	TL-DA	TL	TL-DA
1	91.9	92.5	92.9	92.1	**93.4**	93.0
2	87.3	87.6	87.3	87.4	87.3	**88.9**
3	81.6	81.6	81.2	81.8	81.2	**83.0**
4	97.0	97.0	96.5	96.9	98.3	**98.5**
5	75.3	71.9	70.6	72.6	**78.1**	74.7
6	75.1	76.9	**77.0**	76.9	75.8	76.3
7	69.9	72.7	70.7	64.7	**74.5**	72.7
8	70.7	70.5	71.1	**72.4**	68.9	68.1

FIG. 8B

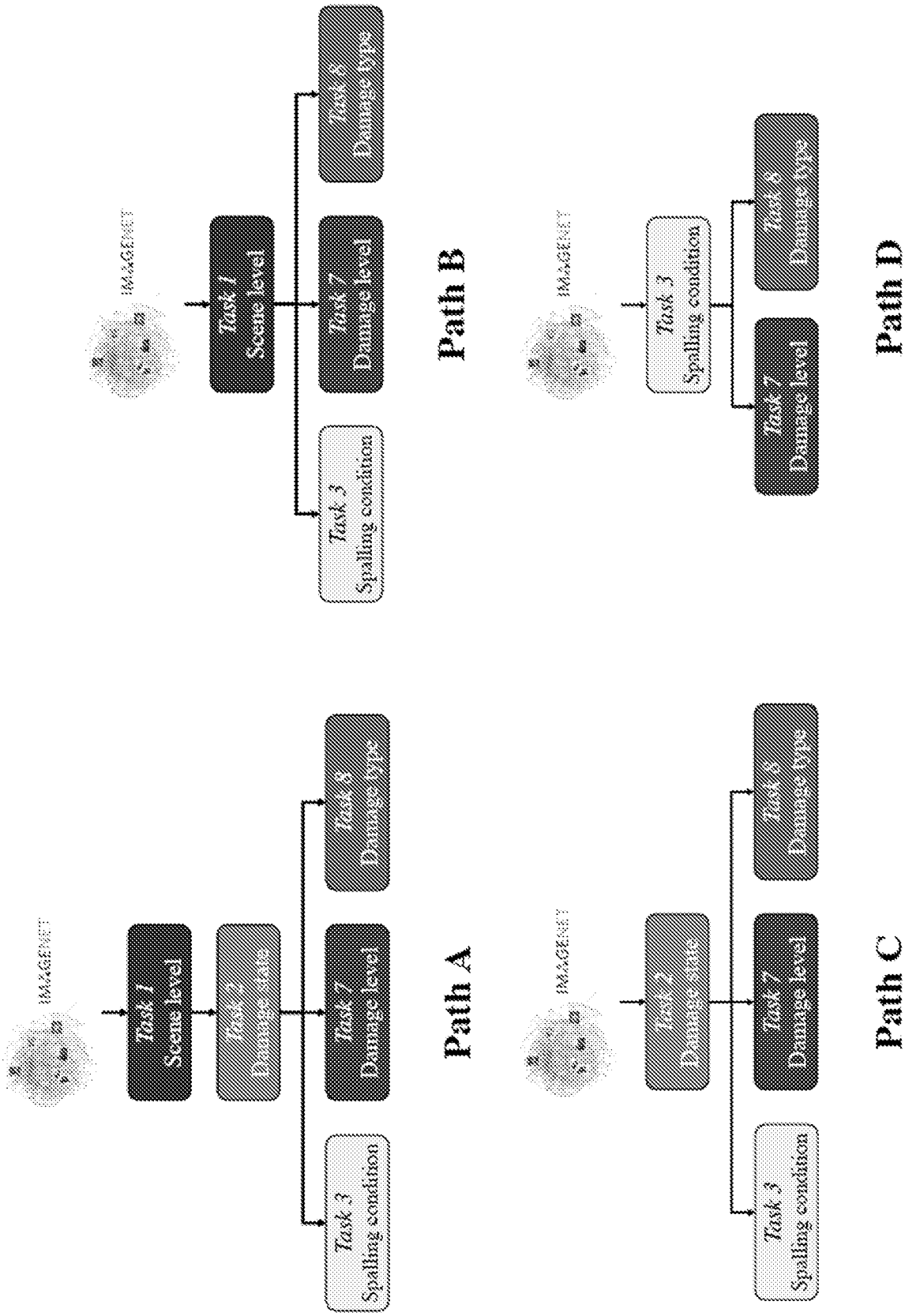

IMAGENET
Task 1
Scene level
Task 2
Damage state
Task 3
Spalling condition
Task 7
Damage level
Task 8
Damage type
Path A
Path B
Path C
Path D

FIG. 9

Table 3: Hierarchical transfer learning results (%)

Path	Source task	Target task	HTL	Over TL
A	ImageNet → 1	2	88.3	+1.0
	ImageNet → 1 → 2	3	83.3	+2.1
	ImageNet → 1 → 2	7	74.1	**-0.4**
	ImageNet → 1 → 2	8	70.9	**+2.0**
B	ImageNet → 1	3	82.6	+1.4
	ImageNet → 1	7	72.1	-2.4
	ImageNet → 1	8	70.5	+1.6
C	ImageNet → 2	3	82.6	+1.4
	ImageNet → 2	7	70.3	-1.4
	ImageNet → 2	8	70.3	+1.4
D	ImageNet → 3	7	74.1	-0.4
	ImageNet → 3	8	69.7	+0.8

FIG. 10

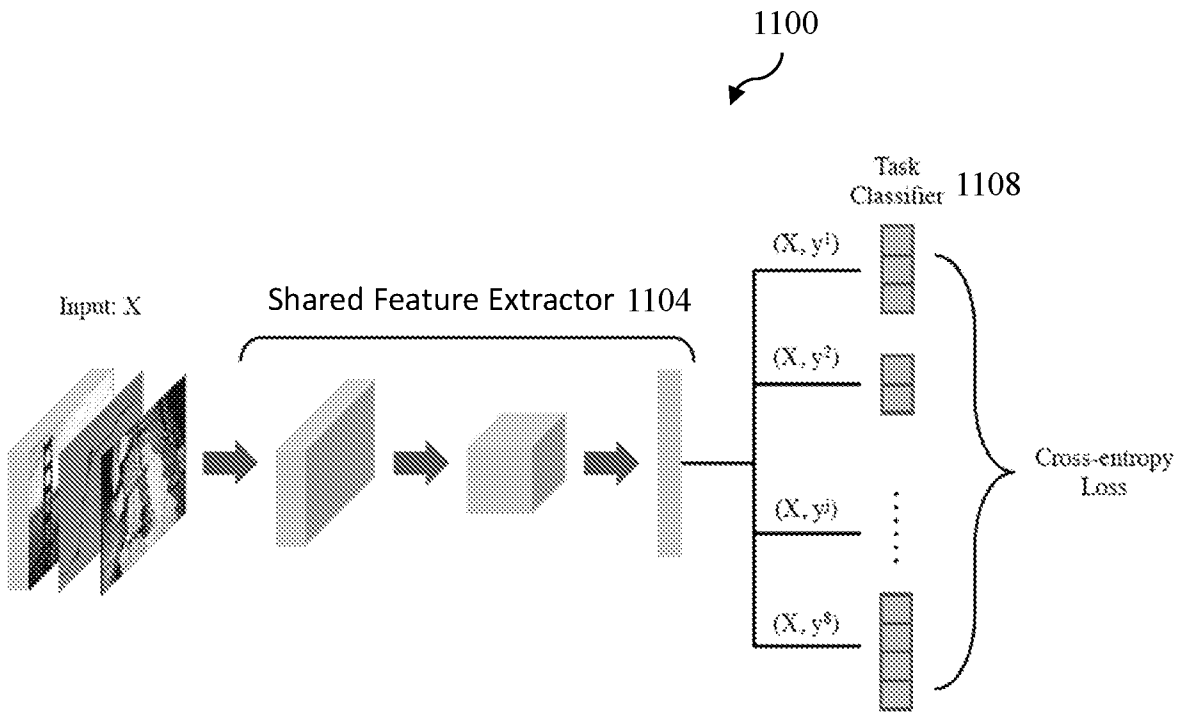


FIG. 11

Table 4: Multi-task transfer learning results (%)

Task	MTL	MTL-DA	over TL	Over TL-DA
1	**94.2**	93.2	+0.8	+0.2
2	**89.7**	88.3	+2.4	-0.6
3	83.4	**85.1**	+2.2	+2.1
4	**99.1**	98.7	+0.8	+0.2
5	**75.3**	74.0	-2.8	-0.7
6	**80.0**	78.6	+4.2	+2.3
7	**76.5**	76.3	+2.0	+3.6
8	73.2	**74.0**	+4.3	+5.9

FIG. 12

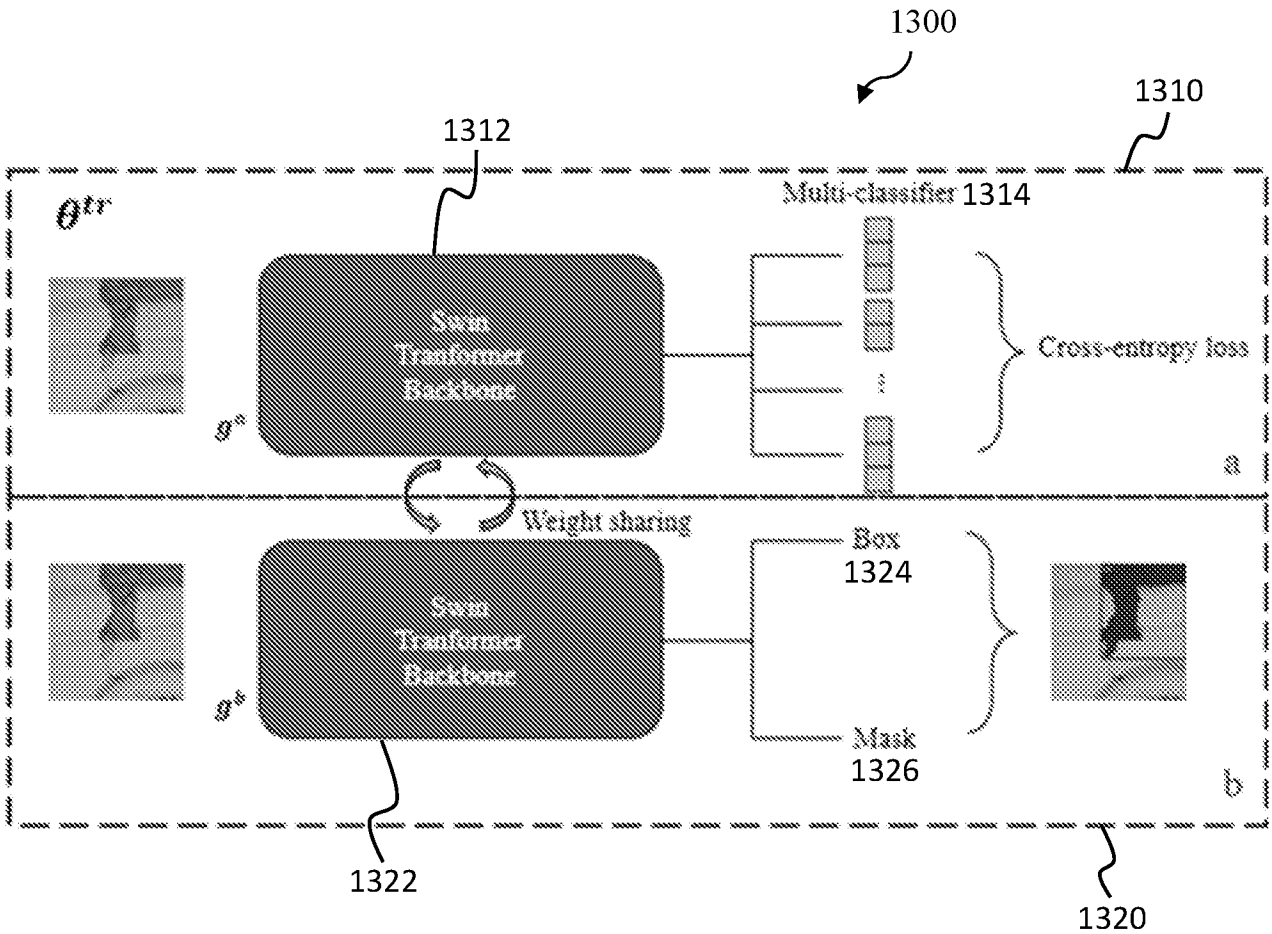


FIG. 13

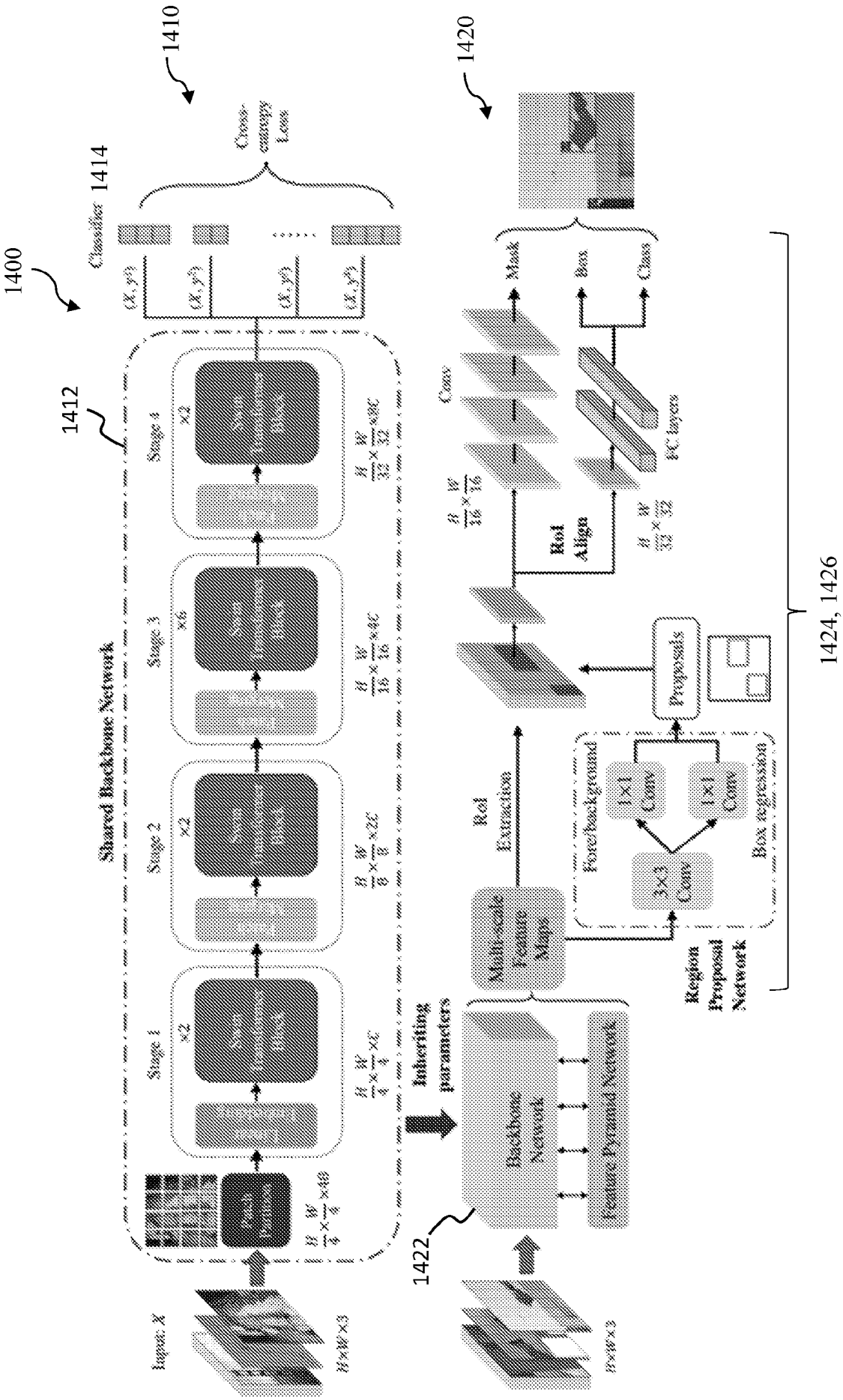

1400
1410
1420
1412
1414
1422
1424, 1426
Cross-entropy Loss
Classifier
Shared Backbone Network
Stage 1
Stage 2
Stage 3
Stage 4
Input X
Inheriting parameters
Backbone Network
Feature Pyramid Network
Multi-scale Feature Maps
RoI Extraction
Region Proposal Network
Foreground
Box regression
3×3 Conv
1×1 Conv
Proposals
RoI Align
Conv
FC layers
Mask
Box
Class

FIG. 14

Table 5: Localization and segmentation results (%)

	Localization		Segmentation	
Metric	HL	MR-CNN	HL	MR-CNN
AP^{50}	**80.6**	72.5	**83.1**	70.3
AP^{75}	**67.4**	55.0	**59.9**	52.0
AP^{S}	**45.5**	43.2	**42.6**	37.5
AP^{M}	**65.2**	55.2	**63.1**	52.6
AP^{L}	**80.2**	59.7	**74.2**	59.4
AR^{50}	**67.9**	59.5	**62.6**	54.8
AR^{75}	**68.2**	59.5	**64.0**	54.8
AR^{S}	**56.2**	50.1	**48.9**	45.2
AR^{M}	**71.4**	66.4	**70.2**	61.6
AR^{L}	**88.9**	71.1	**81.5**	67.2

FIG. 15

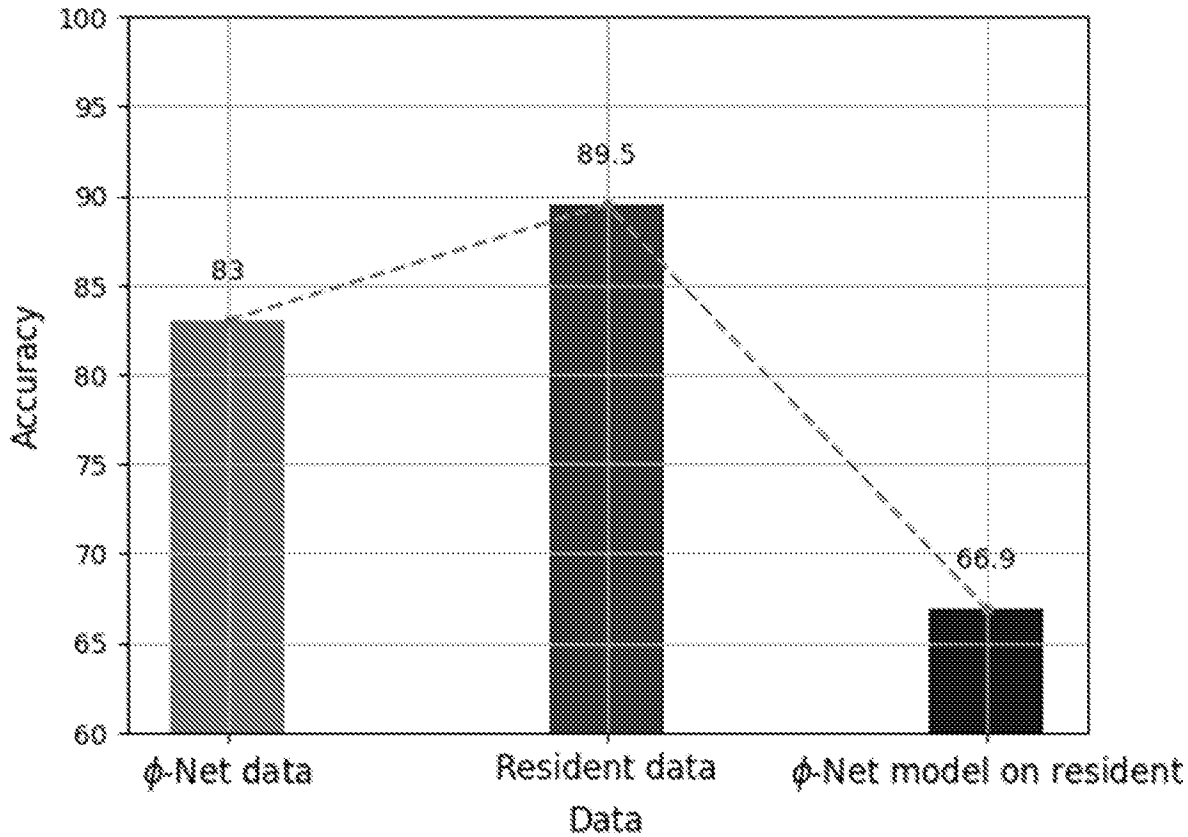


FIG. 16

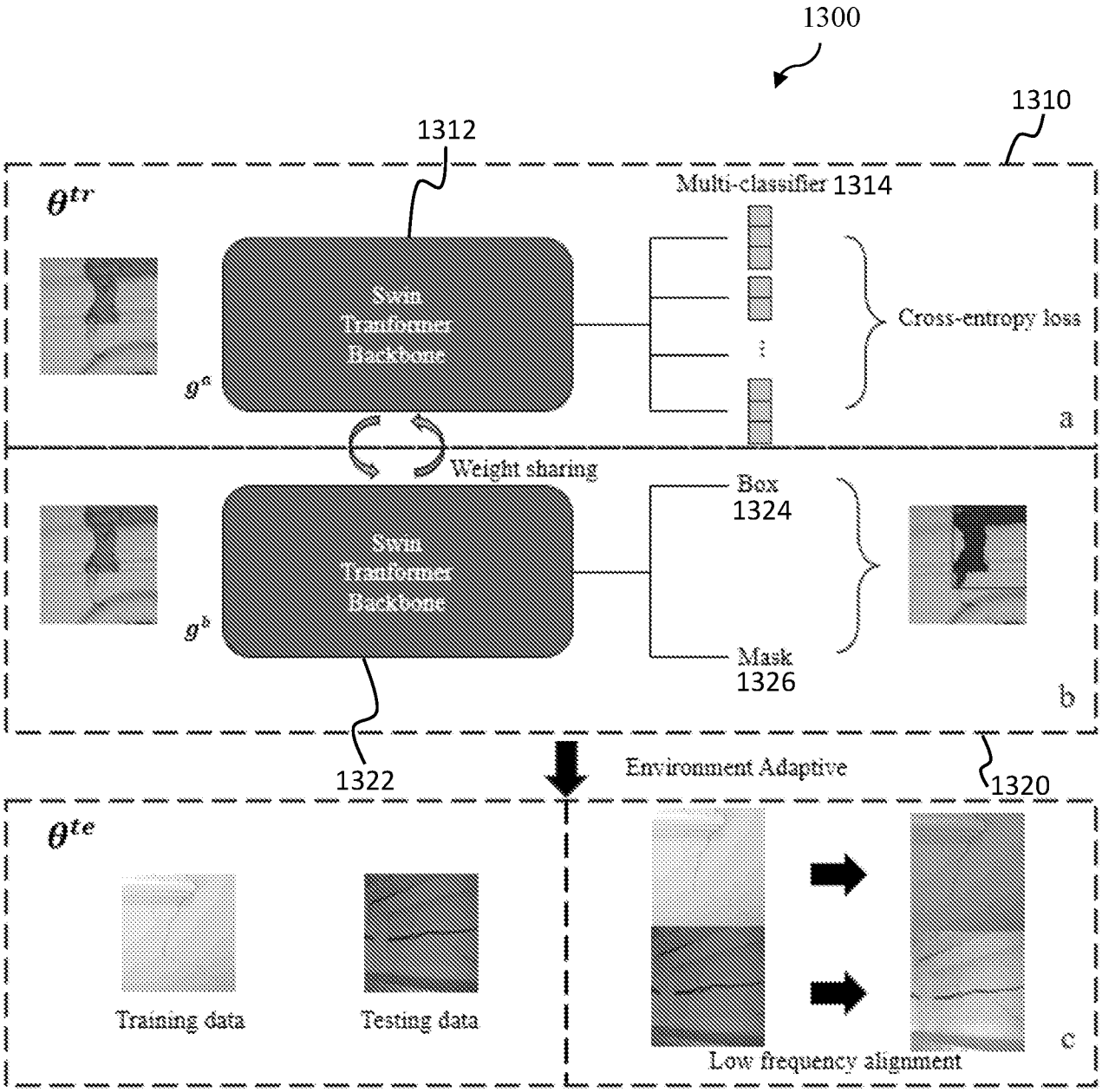

1300
1310
1312
θ^{tr}
Multi-classifier 1314
Swin Tranformer Backbone
g^a
Cross-entropy loss
a
Weight sharing
Box
1324
Swin Tranformer Backbone
g^b
Mask
1326
b
1322
Environment Adaptive
1320
θ^{te}
Training data
Testing data
Low frequency alignment
c

FIG. 17

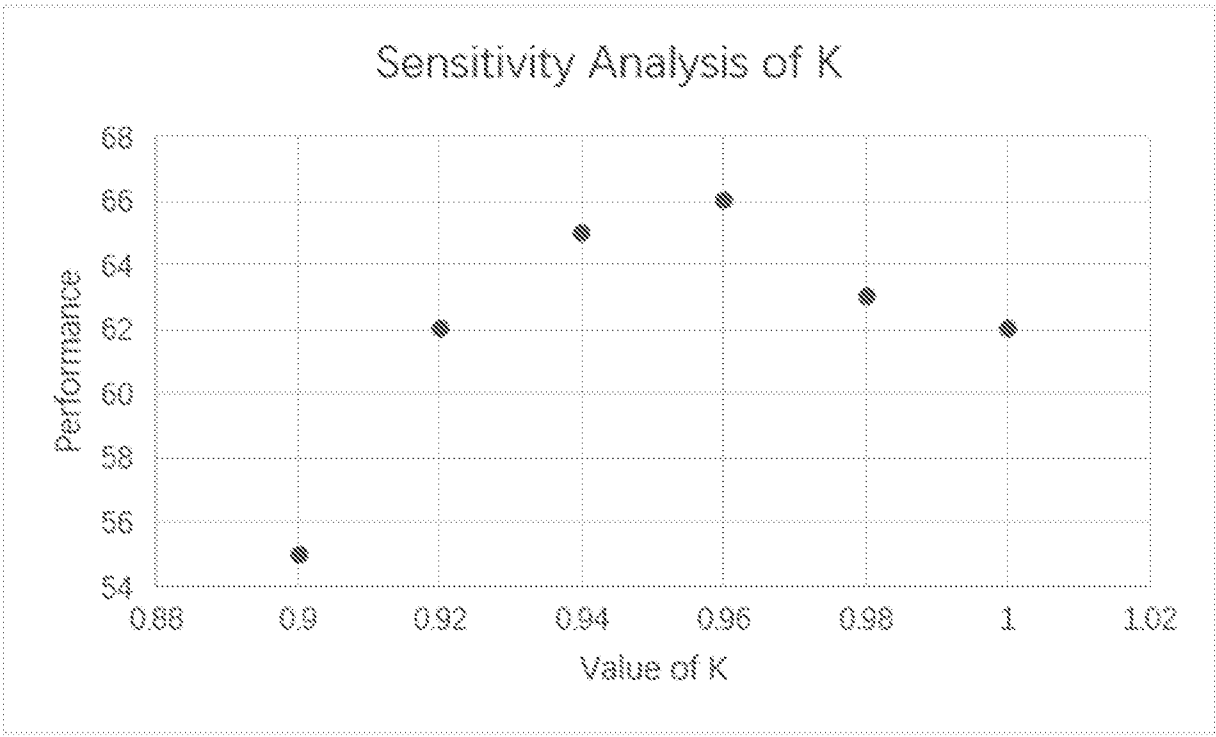

Sensitivity Analysis of K
Performance
68
66
64
62
60
58
56
54
0.88
0.9
0.92
0.94
0.96
0.98
1
1.02
Value of K

FIG. 18

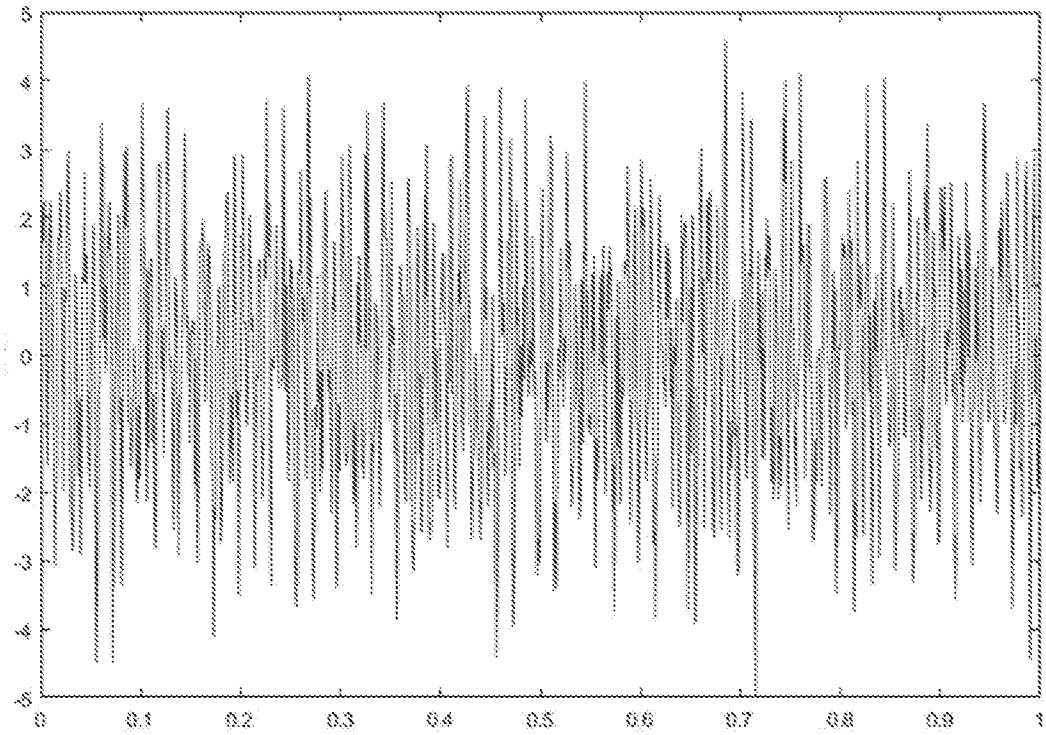


FIG. 19A

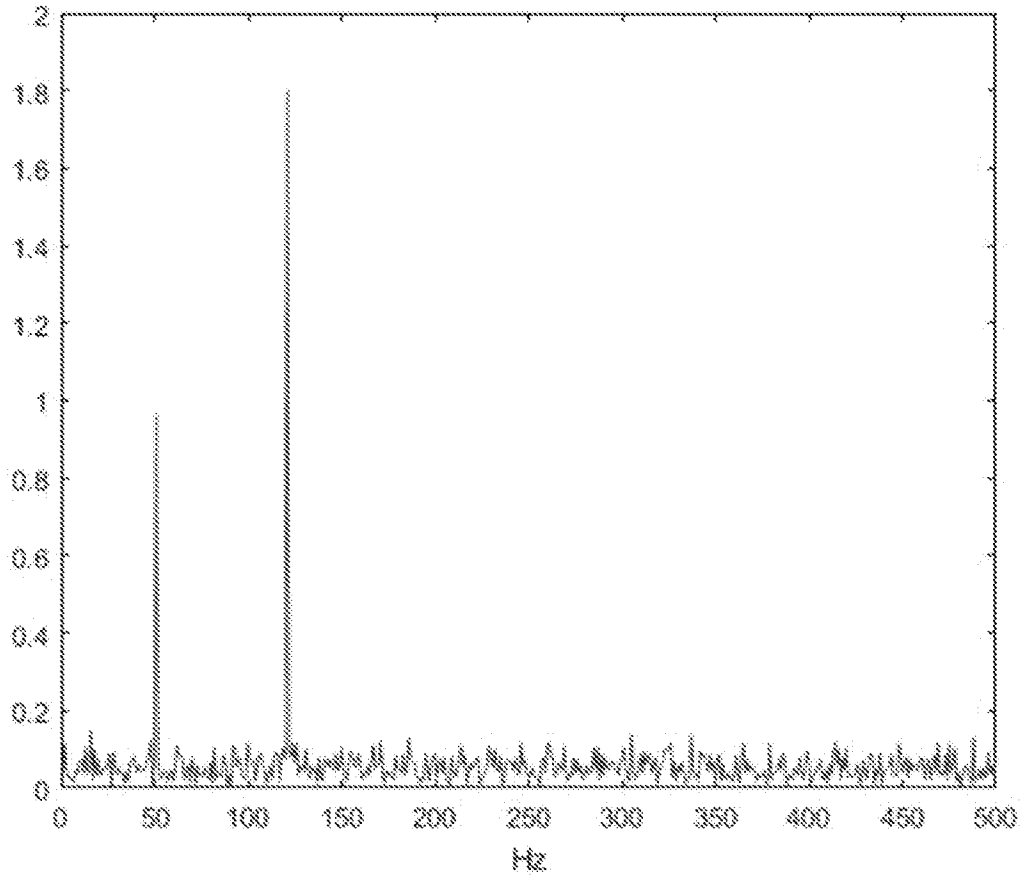


FIG. 19B

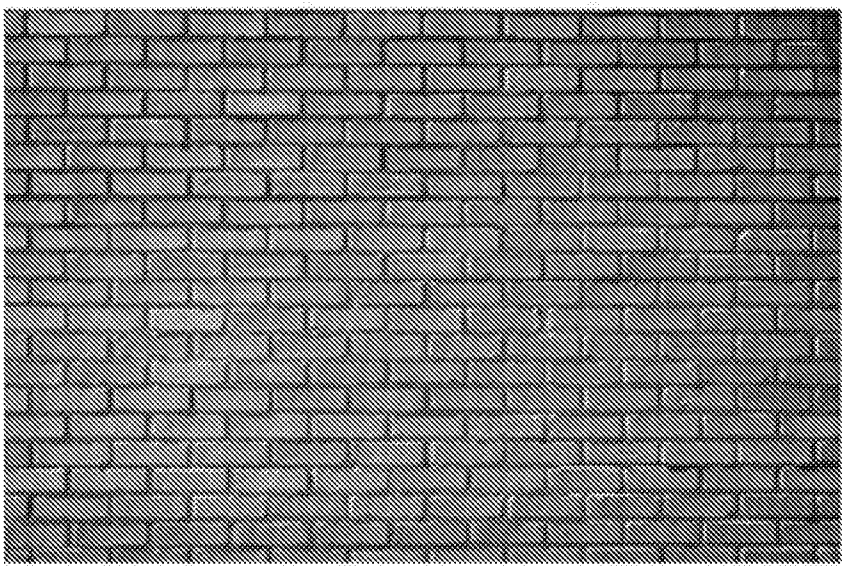


FIG. 20A

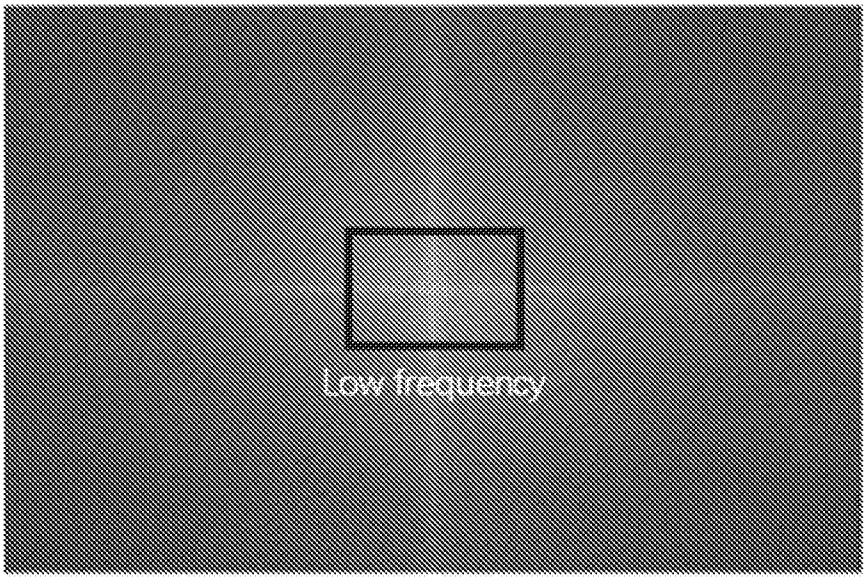


FIG. 20B

Table 6: Comparing the classification performance of different models on a residential dataset (%).

Task	Original model	Environment adaptive
1	60.7	**80.4**
2	55.2	**78.5**
3	66.9	**85.8**
4	71.0	**89.9**
5	75.3	**91.4**
6	69.2	**85.3**
7	60.3	**84.2**
8	65.7	**87.0**

FIG. 21A

Table 7: Comparing the localization performance of different models on a residential dataset (%).

Metric	Original model	Environment adaptive
AP^{50}	50.5	**78.5**
AP^{75}	41.2	**65.4**
AP^{S}	29.8	**43.9**
AP^{M}	48.8	**64.2**
AP^{L}	58.5	**78.3**

FIG. 21B

Table 8: Comparing the segmentation performance of different models on a residential dataset (%).

Metric	Original model	Environment adaptive
AP^{50}	46.0	**75.0**
AP^{75}	38.2	**65.0**
AP^{S}	17.5	**42.5**
AP^{M}	40.2	**62.5**
AP^{L}	51.8	**76.7**

FIG. 21C

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SG2023/050685

A. CLASSIFICATION OF SUBJECT MATTER

G06V 10/88 (2022.01) ***G06N 3/0464 (2023.01)*** *G06Q 50/08 (2012.01)*

According to International Patent Classification (IPC)

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06V, G06N, G06Q

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

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

FAMPAT, Internet: backbone, train, Fourier, neural network, pseudo-labeling, Swin transformer, spectrum, frequency, building, defect, classify, detect and other related terms

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LEE K. ET AL., MultiDefectNet: Multi-Class Defect Detection of Building Facade Based on Deep Convolutional Neural Network. *Substainability*, 23 November 2020, Vol. 12, No. 9785, pages 1-14 [Retrieved on 2023-12-06] <DOI: 10.3390/SU12229785> 3. Research Methodology, 3.1. Overall Architecture, 3.4 Region Proposal Network, 3.5. Fast R-CNN Detector (RoI Pooler), 4.2 Results	1-38
X	NAFI T. I. ET AL., High Accuracy Swin Transformers for Image based Wafer Map Defect Detection. *I. J. Engineering and Manufacturing*, 8 October 2022, No. 5, pages 10-21 [Retrieved on 2023-12-06] <DOI: 10.5815/IJEM.2022.05.02> 1. Introduction, 3.1 Dataset, 3.3.2 Swin Transformer Model, 3.3.3 FNet Transformer Model	1-3, 5, 7-17, 19-21, 23, 25-35, 37

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*Special categories of cited documents:

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

"D" document cited by the applicant in the international application

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
06/01/2024 *(day/month/year)*	22/01/2024 *(day/month/year)*

Name and mailing address of the ISA/SG	Authorized officer
Intellectual Property Office of Singapore 1 Paya Lebar Link, #11-03 PLQ 1, Paya Lebar Quarter Singapore 408533	Yeo Eng Guan (Dr)
Email: pct@ipos.gov.sg	IPOS Customer Service Tel. No.: (+65) 6339 8616

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	AHMED K. ET AL., Performance evaluation of salient object detection techniques. *Multimedia Tools and Applications*, 16 March 2022, No. 81, pages 21741-21777 [Retrieved on 2023-12-06] <DOI: 10.1007/S11042-022-12567-Y> Whole document especially 4.2.4 Frequency and spatial domain analysis, 5.12 Semi-supervised object detection	
A	GUIBAS J. ET AL., ADAPTIVE FOURIER NEURAL OPERATORS: EFFICIENT TOKEN MIXERS FOR TRANSFORMERS. *https://arxiv.org/pdf/2111.13587v1.pdf*, 24 November 2021, pages 1-15 [Retrieved on 2023-12-06] Whole document especially A. 5 Classification, 5.1 ImageNet-1K Inpainting	