

Methods and Systems with Additive AI Models

This application is a continuation-in-part of U.S. utility application serial number 17/134,990 filed December 28, 2020 which is a continuation of International application PCT/US20/66580 filed December 22, 2020 entitled “Efficient artificial intelligence analysis of images with combined predictive modeling” by inventors Seth Wallack, Ariel Ayaviri Omonte, Ruben Venegas, Yuan-Ching Spencer Teng and Pratheev Sabaratnam Sreetharan which claims the benefit of and priority to U.S. provisional applications serial numbers 62/954,046 filed December 27, 2019 and to 62/980,669 filed February 24, 2020, both entitled “Efficient Artificial Intelligence Analysis of Images” by inventors Seth Wallack, Ariel Ayaviri Omonte and Ruben Venegas; and U.S. provisional application serial number 63/083,422 filed September 25, 2020 entitled “Efficient artificial intelligence analysis of images with combined predictive modeling” by inventors Seth Wallack, Ariel Ayaviri Omonte, Ruben Venegas, Yuan-Ching Spencer Teng and Pratheev Sabaratnam Sreetharan, and claims the benefit of U.S. provisional application serial number 63/395,525 filed August 5, 2022, inventors Seth Wallack and Eric Goldman, entitled, “Additive AI classifiers”, each of which is hereby incorporated by reference herein in its entirety.

Background

Artificial intelligence (AI) processors, e.g., trained neural networks are useful to process radiologic images of animals to determine probabilities that the imaged animals have certain conditions. Typically, separate AI processors are used to evaluate respective body regions (e.g., thorax, abdomen, shoulder, fore limbs, hind limbs, etc.) and/or particular orientations (e.g., ventral dorsal (VD) view, lateral view, etc.) of each such body region. A specific AI processor determines for a respective body region and/or orientation, probabilities that particular conditions exist with respect to the body region in question. Each such AI processor includes a large number of trained models to evaluate respective conditions or organs within the imaged region. For example, with respect to a lateral view of an animal’s thorax, an AI processor employs different models to determine probabilities that the animal has certain conditions relating to the lungs, such as perihilar infiltrate, pneumonia, bronchitis, pulmonary nodules, etc.

The amount of processing that is performed by each such AI processor, and the amount of time that is needed to complete such processing is extensive. The task either

requires (1) a manual identification and cropping of each image to define a particular body region and orientation prior to the image being evaluated by a specific AI processor or (2) feeding the images into each AI processor for evaluation. Unlike human radiology in which radiologic studies are limited to specific areas, veterinary radiology routinely includes multiple unlabeled images, with multiple body regions of unknown orientation, within a single study.

In a conventional workflow for processing a radiologic image of an animal, the system assumes that a user-identified body region is contained in the image. The user-identified image is then sent to specific AI processors that, for example, use machine learning models to evaluate the probability of the presence of a medical condition for that specific body region. However, requiring the user to identify body region creates friction in the conventional workflow and leads to errors if the identified body region is incorrect or if multiple regions are contained in the image. Additionally, the conventional workflow becomes inefficient (or breaks down) when images without user identification of body region are sent to the system. When this occurs, the conventional workflow is inefficient because unidentified images are sent to a large number of AI processors which are not specific to the imaged body region. Further, the conventional workflow is prone to false results because incorrect region identification results in images being sent to AI processors that are configured to evaluate different body regions.

The conventional workflow for analyzing diagnostic features of a radiograph using AI and preparing a report based on AI model diagnostic results in an exponential number of possible output reports. An AI model diagnostic result provides either a normal or an abnormal determination with respect to a particular condition. In some AI models, a determination of the severity of a particular condition e.g. normal, minimal, mild, moderate, or severe, is also provided. A collection of AI model diagnostic results determine which report is to be selected from premade report templates. The process of creating and choosing a single report template from a collection of AI model diagnostic results scales exponentially with the number of AI models. Six different AI model normal/abnormal diagnostic results require 64 different report templates (two raised to the sixth power). Ten models require 1,024 templates, and 16 models require 65,536 templates. AI models that detect severity scale even more poorly, for example 16 severity detection models with 5 possible severities each would require over 150 billion templates. Therefore, a manually created report for each combination of AI model diagnostic results does not scale well to a large number of AI models being interpreted together.

Therefore, there exists a need for a novel system which has several fully automated stages of image preprocessing and image analysis, including determining whether the received image includes a particular body region in a particular orientation (lateral view, etc.); cropping the image appropriately; creating one or more sub-images from an original
5 image that contains more than one body region or region of interest; labeling the original image and any sub-images created; and evaluating the cropped image and sub-images against targeted AI models. Further, there exists a need for a novel system which analyzes and provides a diagnostic radiologist report based on a large number of test results, including but not limited to, AI model results.

10 AI models or classifiers in machine learning undergo continual training to improve model performance. The best practices for AI model deployment involve continual training which includes retraining the currently deployed AI model. The retraining is based on one of more of these common parameters such as AI performance based, trigger based on data changes or training on demand. The current standard of industry for retraining of AI models
15 has an objective of replacing the currently trained model with a new, and “improved” AI model. The current AI technology is based on single model results in a production environment therefore has developed the concept of retrain and replace.

The problem with the current approach is that replacing a current AI model with a newly trained AI model works under the assumption that the improvement in model
20 performance is reducing unwanted “noise” or false positive data. The data detected as unwanted “noise” in the current AI model are removed and replaced with the retrained AI model. However, the assumption that the “noise” is not valuable information in overall system performance is incorrect. The “noise” being detected by the current model classifier is useful to distinguish data points with similar but not identical characteristics. Therefore, by
25 replacing the current AI model data are lost.

Therefore, there is a need for a new approach to continual training of AI models that incorporates both old and new AI models.

Summary

An aspect of the inventions described herein provides a method for obtaining an
30 additive AI result from a digital file, the method including: processing the digital file by a first artificial intelligence (AI) classifier and at least one second AI classifier thereby obtaining a first evaluation result and at least one second evaluation result respectively; directing the first evaluation result and the at least one second evaluation result to at least one

synthesis processor; and comparing the first evaluation result and the at least one second evaluation result to at least one dataset cluster thereby obtaining the additive AI result. The terms AI model and AI classifier are used interchangeably and are defined as a type of machine learning algorithm used to assign a class label to a data input.

5 An embodiment of the method further includes measuring distance from the additive AI result to an exemplar result from the dataset cluster to obtain an additive AI cluster identification. In an embodiment of the method, the dataset cluster further includes matched written templates. An embodiment of the method further includes assembling the additive AI cluster identification and the matched written templates to obtain a report. An embodiment of
10 the method further includes displaying the report to a user.

 In an embodiment of the method, the second AI classifier is a derivative of the first AI classifier. In an embodiment of the method, the second AI classifier is trained using at least a portion of data used for training the first AI classifier. For example the second classifier is trained using at least 99%, 95%, 90%, 85%, 80%, 75%, 70%, 66%, 60%, 55%, 50%, 45%,
15 40%, 35%, 30%, 25%, 20%, 15%, 10%, or 5% of data used for training the first AI classifier. In an embodiment of the method, the second AI classifier is related to first AI classifier. In some embodiments, the first AI classifier is a comprehensive classifier. In some embodiments, the second AI classifier is a specific classifier. In alternative embodiments, the first AI classifier is a specific classifier and the second AI classifier is a comprehensive
20 classifier.

 An embodiment of the method further includes repeating the steps of directing and comparing for a series of daisy chained AI classifiers. An embodiment of the method further includes comparing the first evaluation result to the second evaluation result for training the first and the second AI classifier or for comparing AI result and testing expected
25 performance.

 An embodiment of the method further includes adding the first evaluation result and the second evaluation result to a results database. An embodiment of the method further includes prior to processing, acquiring the digital file. An embodiment of the method further includes prior to processing, converting an analog file into the digital file. An embodiment of
30 the method further includes prior to processing, classifying the digital file by performing at least one of: labelling, cropping, editing, and orientating the digital file.

 An embodiment of the method further includes adjusting the AI model result in a heuristic manner by applying a mathematical formula to the AI model result including at least

one of: addition, subtraction, multiplication, division or any other standard mathematical formula.

An aspect of the invention described herein provides a system programmed to obtain an additive AI result by any of the methods described herein, the system including: at least one first AI processor; at least one derivative AI processor that is derived from the first AI processor; and an output device.

An embodiment of the system further includes at least one database library. An embodiment of the system further includes a user interface. In some embodiments, the AI results from one derivative classifier for only one first AI classifier. In some embodiments, more than one derivative classifier for only one first AI classifier is included in the results. In some embodiments, one derivative classifier for more than one first AI classifier is included in the results. In some embodiments, more than one derivative classifier for more than one first AI classifier is included in the results. In some embodiments, one derivative classifier for only some of the first AI classifiers and more than one derivative classifier for some of the first AI classifiers is included in the results.

Brief description of drawings

FIG. 1 is a schematic drawing of a conventional workflow for processing a radiologic image 102 of an animal. As commonly represented in the veterinary field, the image 102 does not indicate portion(s) of the animal. The image 102 is processed by each of a large number of AI processors 104a-104l to determine if that body region is present on the image and the probabilities that the animal represented in the image 102 has certain conditions. Each of the AI processors 104a-104l evaluate the image 102 by comparing the image to one or more machine learning models that have each been trained to determine a probability that the animal has a particular condition.

FIG. 2 is a schematic drawing of an embodiment of the system or the methods described herein. A radiologic image pre-processor 106 is deployed to pre-process the image 102 to generate one or more sub-images 108 that each corresponds to a particular view of a specific body region. Three sub-images 108a-c were generated, with one sub-image 108a identified and cropped as a lateral view of a thorax of the animal, second sub-image 108b identified and cropped as a lateral view of the animal's abdomen, and third sub-image 108c identified and cropped as a lateral view of the animal's pelvis.

As indicated, the sub-image 108a is processed only by the lateral thorax AI processor 104a, the sub-image 108b is processed only by the lateral abdomen AI processor 104c, and

the sub-image 108c is processed only by the lateral pelvis AI processor 104k. In some embodiments, the sub-images 108 is tagged to identify the body region and/or view that sub-image 108 represents.

FIG. 3 is a description of a set of computer operations performed by an embodiment of the system or the method of the invention herein for the novel workflow described herein. An image 302 is processed using the radiologic image pre-processor 106 followed by a subset of AI processors 104 corresponding to identified body regions/views. Cropped images 304a, 304b of respective body regions/views that are identified by the system are shown. The total time taken by the radiologic image pre-processor to determine that the image 302 represented both a "lateral thorax" image and a "lateral abdomen" image, as reflected by the time stamps for the log entries corresponding to the bracket 306, was twenty-four seconds.

FIG. 4 is a set of conventional single condition base organ findings 401 to 407 for lungs in a radiograph followed by combinations of at least two single condition base organ findings. The permutations and combinations of seven single condition base organ findings result in exponential number of report templates.

FIG. 5A- FIG. 5F are a set of base organ findings for lungs classified based on severity as normal, minimal, mild, moderate, and severe and displayed as separate AI model results templates. The boxes 501 to 557 represent a single line item in a specific AI report template. The single line item is selected based on each AI model result template matching the finding listed under heading "Code".

FIG. 6 is a collection of individual binary models (or a library of AI models) which are deployed for analyzing radiologic images to obtain a probability result of the radiologic image being negative or positive for the condition or classification.

FIG. 7 is a lateral thoracic radiograph of a dog, the image having been preprocessed, cropped, labeled and identified. The radiograph image is analyzed by the library of binary AI models displayed in FIG. 6.

FIG. 8 is a screenshot of results of a single binary AI model obtained by analyzing a series of lateral radiologic images similar to the image of FIG. 7 through a specific binary AI model, for example, a bronchitis AI model.

FIG. 9A- FIG. 9E are a set of screenshots showing each AI model result of a radiologic image. The visual collection of each individual AI model result per image and the AI model result mean for all images were evaluated for that specific case. The mean evaluation result for each model is created by assembling the individual image evaluation results, and is displayed at the top of the screen in FIG. 9A. Each individual image and the AI

model result for that image are displayed in FIG 9B-FIG. 9E. The time stamp 901 in FIG. 9A shows that the AI analysis was completed in less than three minutes. FIG. 9B shows results for individual AI models such as perihilar infiltrate, pneumonia, bronchitis, interstitial, diseased lungs, hypoplastic trachea, cardiomegaly, pulmonary nodules, and pleural effusion.

5 For each AI model a label identifies the image as “normal” 902 or “abnormal” 903. Further, a probability 904 of the image being “normal” or “abnormal” for the AI model single condition is provided. FIG. 9C shows four images which are obtained by classifying and cropping a single radiologic image. The time stamps 905-908 show that the AI analysis was completed in less than two minutes. FIG. 9D and FIG. 9E show results for each AI model for the

10 radiologic image which includes the label, the probability and the view (909) of the radiologic image, for example, lateral, dorsal, anteroposterior, posteroanterior, ventrodorsal, dorsoventral, etc.

FIG. 10 is a screenshot of an AI case result displayed in JavaScript Object Notation (JSON) format. The JSON format facilitates copying the mean evaluation results for all

15 models in a case which may be transferred to an AI Evaluation Tester for testing to evaluate the mean evaluation results by comparing to a cluster result.

FIG. 11 is a screenshot of a graphical user interface which allows a user to create a K-Means cluster. The user assigns a name 1101 for the new cluster under “Code”. The user selects various parameters to create a cluster. The user chooses a start Case date 1102 and an

20 end Case date 1103 to select the cases. The user chooses a start Case ID 1104 and an end Case ID 1105 to select the cases. The user chooses a maximum number of cases 1106 that are to be included in the cluster. The user chooses species 1107 such as dog, cat, dog or cat, human, etc. for the cases to be included in the cluster. The user selects specific diagnostic modalities 1108 such as X-ray, CT, MRI, blood analysis, urinalysis, etc. to be included in

25 creating the cluster. The user specifies separating the evaluation results into a specific number of clusters. The number of clusters range from a minimum of one cluster to a maximum number of clusters limited only by total number of cases entered into the cluster.

FIG. 12 is a screenshot of AI cluster results listed as a numerical table. The far left column 1201 is the case ID, the next nine columns are the mean evaluation results 1202 for

30 each binary model for the specific case ID, the next column is the cluster label or cluster location 1203 which includes the specific case based on the collection of evaluation results, the next four columns are the cluster coordinates and centroid coordinates, the last number is the case ID 1204 of the centroid or center of that specific cluster. The radiologist report for the best matched case ID is obtained. This radiologist report is then used to generate the

report for the new AI case. This process allows for infinite scalability in terms of the number of AI models incorporated compared to the conventional semi-manual process of report creation.

FIG. 13 is an example of a clustering graph. The clustering graph is created by
5 dividing the mean evaluation results into multiple different clusters depending on user defined parameters 1102-1108. This example clustering graph is divided in 180 different clusters each represented by collection nearby of dots of a single color plotted on the graph.

FIG. 14 is a screenshot of user interface showing AI cluster models generated based on user defined parameters 1102-1108. The first column from the left shows the cluster ID
10 1401, the second column shows assigned name of the cluster model 1402, the third column shows the number of different clusters 1403 into which the AI data have been divided, fourth column shows the body region 1404 that has been evaluated based on cluster data results.

FIG. 15 is a screenshot of a user interface showing screening evaluation configuration. The user interface allows assigning a specific “cluster model” 1502 to a
15 specific “screening evaluation configuration” name 1501. The status 1503 of the screening evaluation configuration provides additional data about the configuration such as whether the configuration is in live, testing or draft mode. The live mode is for production and the testing mode is for development.

FIG. 16A -FIG. 16C are a set of screenshots of a user interface showing the details for
20 a specific cluster model. FIG. 16A shows a user interface displaying data for a cluster model 1601 Thorax 97. The AI evaluation classifier types 1602 included in the cluster are listed. The species or a collection of species 1603 specific for the cluster model are displayed. The maximum number of cases 1604 with evaluation results used to generate the cluster are displayed. The user interface shows the start and end dates 1605 for cases used to create the
25 cluster. A link 1606 to the comma separated value (CSV) file of FIG. 12 showing the cluster in numerical table format is displayed. A portion of sub-clusters 1608 created from the parameters 1602-1605 are listed. The total number of sub-clusters 1609 created for this cluster group are displayed. For each sub-cluster a centroid case ID 1610 is displayed. A link to the log 1607 for building the cluster is displayed. FIG. 16B is a screenshot of the log
30 created for cluster model 1601 Thorax 97. FIG. 16C is a screenshot of a portion of AI evaluation models including vertebral heart score, perihilar infiltrate, pneumonia, bronchitis, interstitial, and diseased lungs.

FIG. 17A- FIG. 17D are a set of screenshots of a user interface for AI Evaluation Tester. FIG. 17A shows user interface (AI Eval Tester) in which the values of mean

evaluation results for all models in JSON format of FIG. 10 are imported 1701 to analyze the closest matched case/ exemplar result match in a cluster using K-means clustering from a case cluster made from an AI dataset. FIG. 17B shows the values of mean evaluation results for all models in JSON format of FIG. 10 being imported into the AI Evaluation tester. FIG. 5 17C and FIG. 17D show the evaluation results which have been imported for the specific case. FIG. 17D shows screening evaluation type 1702 and the cluster model 1703 associated with the screening evaluation type being selected by the user. By clicking test 1704, the evaluation results displayed in FIG. 10 are analyzed and assigned to the closest matched case/ exemplar result match in a cluster. The closest radiologist report, top ranking radiologist 10 sentences and centroid radiologist report for the exemplar result match cluster are collected and displayed.

FIG. 18A- FIG. 18E are a set of screenshots of a user interface. FIG. 18A and FIG. 18B are a set of screenshots of the user interface which shows the result displayed after clicking test 1704 on the AI Evaluation tester. The diagnosis and conclusory findings 1801 15 from radiologist report that are closest to the evaluation results based on previously created cluster results are displayed. The evaluation findings 1802 are selected from radiologist reports in the cluster of the evaluation results and filtered based on the prevalence of a specific sentence in the findings section of the specific cluster. The recommendations 1803 from the radiologist report in the cluster are selected based on prevalence of each sentence or 20 a similar sentence in the recommendations section of this cluster. The interface 1804 shows the radiologist report of the cluster and the interface 1805 shows the radiologist report of the centroid of the cluster. FIG. 18C is a screenshot of a user interface which lists the ranking of the sentences in a radiology report based on the specific cluster results. The sentences include conclusory sentences 1806, findings sentences 1807, and recommendations sentences 1808. 25 FIG. 18D and FIG 18E are a screenshot of a user interface which allows the user to edit the radiology report by editing the findings section 1809, the conclusion section 1810 or the recommendation section 1811 by adding or removing specific sentences.

FIG. 19 is a radiologist report for the closest match dataset case, which is used to generate the radiology report for the new case. The AI Evaluation tester displays the closest 30 radiologist report to the current AI evaluation results based on similarity of the evaluation results between the new image AI evaluation results and the AI evaluation results within the cluster and the radiologist report from the centroid of the selected cluster.

FIG. 20A and FIG. 20B are a set of radiographs. FIG. 20A is the newly received radiograph being analyzed and FIG. 20B is the radiograph that is selected by the results of the

AI evaluation as the closest match based on the cluster model. The cluster match is based on AI evaluation results rather than image match results.

FIG. 21A and FIG. 21B are a set of schematic drawings of the components in an AI radiograph processing unit. FIG. 21 A is a schematic drawing showing that the radiograph machine 2101 sends the radiologic image to a desktop application 2102 which directs the image to a web application 2103. The computer vision application 2104 and the web application directs the images to an image web application which directs the image to AI evaluation 2105. FIG. 21B is a schematic drawing of components in an image match AI processing. The images uploaded in Local Interface to Online Network (LION) 2106 in a veterinary clinic are directed to a VetConsole 2107 which autorotates and auto-crops the images to obtain sub-images. The sub-images are directed to three locations. The first location is to the VetAI console 2108 to classify the image. The second location is to image match console 2109 to add the sub-images with reports to image match database. The third location is to the image database 2110 which stores new images and the corresponding case ID numbers. The image match console 2109 directs the images to refined image match console 2111 or VetImage Editor console 2112 for further processing.

FIG. 22A and FIG. 22B are a set of schematic drawings of the server architecture for image matching. FIG. 22A is a schematic drawing of the server architecture being currently used in AI radiograph analysis. FIG. 22B is a schematic drawing of the server architecture for AI radiograph analysis including pre-processing the radiologic images, analyzing the images using AI diagnostic processors and preparing reports based on clustering results. The image from the PC 2201 is directed to a NGINX load balancing server 2202 which directs the image to V2 cloud platform 2203. The image is then directed to the image match server 2204, the VetImages server 2205 and the database Microsoft SQL server 2207. The VetImages server direct the image to VetAI server 2206, the database Microsoft SQL server 2207 and the datastore server 2208.

FIG. 23A- FIG. 23F are a series of schematic drawings of Artificial intelligence autocropping and evaluation workflow for an image obtained for a subject. The workflow is classified into six columns based on the platform used to accomplish the task such as, the clinic, the V2 end user application, VetImages web application, VetConsole python scripting application, VetAI machine learning application, ImageMatch orientation python application and ImageMatch validation python application. Further, the tasks are shaded a different shade of grey based on the processor that accomplishes the task, such as sub-image processor, evaluation processor and synthesis processor. The V2 application is an end user application in

which a user interacts with the application and uploads the images to be analyzed. The VetImages application processes the images to generate AI result or AI report or evaluation result. The VetConsole is a python scripting app which improves the image quality and processes images in batches. The VetAI is a machine learning application to create AI models and evaluate images which are entered in the system. ImageMatch orientation is a python app which conducts search for correctly oriented images in its database similar to the inputted image. ImageMatch validation is a python app which conducts search for correct classified images in its database similar to the entered image. The sub-image processor accomplishes the tasks 2301-2332 listed in FIG. 23A-FIG.23C. The evaluation processor conducts the tasks 2333-2346, 2356 and 2357 listed in FIG. 23D and a portion of FIG. 23E and FIG. 23F. The synthesis processor performs the tasks 2347-2355 and 2358-2363 listed in FIG. 23F and a portion of FIG. 23E.

FIG. 24 is a schematic of the current model showing image classifiers being replaced sequentially, with the oldest model at the top, the most recent model at the bottom.

FIG. 25 is a new model shown by the claimed approach where derivative image classifiers are used together rather than replaced, where *I* indicates the same image classifier and the number or *N* indicates the derivative classifier with *N* representing any positive integer.

FIG. 26 shows particular embodiments where text versions can be associated with daisy chained derivative classifiers creating an *n*:1 relationship.

FIG. 27 shows particular embodiments where multiple text versions can be associated with daisy chained derivative classifiers creating an *n*:*n* relationship.

FIG. 28 shows visual examples of both derivative (same letter and subscript number) and non-derivative (same letter but different subscript letter) data together. The database holds all the training data with the relationships. This system allows for data inputs (in this case *I*, *T* and *OI*); single or clustered, AI or non-AI data, derivative or non-derivative data, to obtain an exemplar result from the database.

FIG 29A shows a radiograph of a cat thorax that has minimal lung pathology. GLC result of 0.68; GLC2 result of 0.69. These results together are used as a check and balance system to confirm each classifier result.

FIG. 29B shows a radiograph of a cat thorax that has moderate lung pathology, specifically a moderate bronchial pattern. Both GLC and its derivative classifier, GLC2, are trained to identify a bronchial pattern in a radiograph. GLC result of 0.85; GLC2 result of 0.78. These results together are used as a check and balance system to confirm the classifier result is a true positive.

Detailed description

An aspect of the invention described herein provides a method for analyzing a diagnostic radiologic image or an image of a subject, the method including: processing automatically the radiologic image of the subject using a processor for classifying the image to one or more body regions or body regions and orienting and cropping a classified image to obtain at least one oriented, cropped and labeled sub-image for each body region that is automatically classified; directing the sub-image to at least one artificial intelligence processor; and evaluating the sub-image by the artificial intelligence processor thereby analyzing the radiologic image of the subject.

An embodiment of the method further includes using the artificial intelligence processor for assessing the sub-image for body regions and for a presence of a medical condition. Body regions are for example: thorax, abdomen, forelimbs, hindlimbs, etc. An embodiment of the method further includes using the artificial intelligence processor for diagnosing the medical condition from the sub-image. An embodiment of the method further includes using artificial intelligence processor for assessing the sub-image for a positioning of the subject. An embodiment of the method further includes rectifying the positioning of the subject to proper positioning.

In an embodiment of the method, the processor automatically rapidly processing the radiologic image to obtain the sub-image. In an embodiment of the method, the processor processing the radiologic image to obtain the sub-image in: less than about one minute, less than about 30 seconds, less than about 20 seconds, less than about 15 seconds, less than about 10 seconds, or less than about 5 seconds. In an embodiment of the method, evaluating further includes comparing the sub-image to a plurality of reference radiologic images in at least one of a plurality of libraries. In an embodiment of the method, the plurality of libraries each includes a respective plurality of the reference radiologic images.

In an embodiment of the method, each of the plurality of libraries include respective plurality of reference radiologic images specific or non-specific to an animal species. An embodiment of the method further includes matching the sub-image to a reference radiologic

image thereby assessing orientation and at least one body region. In an embodiment of the method, the reference radiologic images are oriented in Digital Imaging and Communication in Medicine (DICOM) standard hanging protocol.

5 In an embodiment of the method, cropping further includes isolating a specific body region in the sub-image. An embodiment of the method further includes categorizing the reference radiologic images according to veterinary radiologic standard body region labels. In an embodiment of the method, orienting further includes adjusting the radiologic image to veterinary radiologic standard hanging protocol. In an embodiment of the method, cropping further includes trimming the radiologic sub-images to a standard aspect ratio. In an
10 alternative embodiment of the method, cropping further does not include trimming the radiologic sub-images to a standard aspect ratio. In an embodiment of the method, classifying further includes identifying and labeling body region according to veterinary standard body region labels. In an embodiment of the method, classifying further includes comparing the radiologic image to a library of sample standard radiologic images.

15 An embodiment of the method further includes matching the radiologic image to a sample standard image in the library thereby classifying the radiologic image to one or more body regions. In an embodiment of the method, cropping further includes identifying a boundary in the radiologic image delineating each classified body region. An embodiment of the method further includes prior to classifying, extracting a signature of the radiologic
20 image. In an embodiment of the method, the radiologic image is from a radiology exam selected from: radiographs viz., X-ray, magnetic resonance imaging (MRI), magnetic resonance angiography (MRA), computed tomography (CT), fluoroscopy, mammography, nuclear medicine, Positron emission tomography (PET), and ultrasound. In an embodiment of the method, the radiologic image is a photograph.

25 In an embodiment of the method, the subject is selected from: a mammal, a reptile, a fish, an amphibian, a chordate, and a bird. In an embodiment of the method, the mammal is selected from: dog, cat, rodent, horse, sheep, cow, goat, camel, alpaca, water buffalo, elephant, and human. In an embodiment of the method, the subject is selected from: a pet, a farm animal, a high value zoo animal, a wild animal, and a research animal. An embodiment
30 of the method further includes automatically generating at least one report with evaluation of the sub-image by the artificial intelligence processor.

An aspect of the invention described herein provides a system for analyzing radiologic images of a subject, the system including: a receiver to receive a radiologic image of the subject; at least one processor to automatically run an image identification and

processing algorithm to identify, crop, orient and label at least one body region in the image to obtain a sub-image; at least one artificial intelligence processor to evaluate the sub-image; and a device to display the sub-image and evaluated artificial intelligence result.

In an embodiment of the system, the processor automatically rapidly processes the radiologic image to obtain the sub-image. In an embodiment of the system, the processor processes the radiologic image to obtain the labeled image in: less than one minute, less than 30 seconds, less than 20 seconds, less than 15 seconds, less than 10 seconds, or less than 5 seconds. An embodiment of the system further includes a library of standard radiologic images. In an embodiment of the system, the standard radiologic images comply with veterinary norms for hanging protocol and body region labels.

An aspect of the invention described herein provides a method for rapidly and automatically preparing radiologic images of a subject for display, the method including: processing an unprocessed radiologic image of the subject using a processor to algorithmically classify the image to one or more separate body region categories, by automatically cropping, extracting a signature and comparing a cropped, oriented image signature to a database of signatures of images of known orientation and body regions to obtain a best match orientation and body region labeling; and, presenting each prepared body region labeled image on a display device and for analysis.

An aspect of the invention described herein provides an improvement in a veterinary radiograph diagnostic image analyzer, the improvement including running a rapid algorithm with a processor that pre-processes a radiograph image of a subject to automatically identify one or more body regions in the image; the processor further functions to perform at least one of: automatically creating a separate sub-image for each identified body region, cropping and optionally normalizing an aspect ratio of each sub-image created, automatically labeling each sub-image as a body region, automatically orienting the body region in the sub-image, and the processor further automatically directs the diagnostic sub-image to at least one artificial intelligence processor specific for evaluating cropped, oriented and labeled diagnostic sub-image.

An aspect of the invention described herein provides a method for identifying and diagnosing a presence of a disease or a condition in at least one image of a subject, the method including: classifying the image to one or more body regions, labelling and orientating the image to obtain a classified, labeled and oriented sub-image; directing the sub-image to at least one artificial intelligence (AI) processor to obtain an evaluation result, and comparing the evaluation result to a database with evaluation results and matched written

templates or at least one dataset cluster to obtain at least one cluster result; measuring the distance between the cluster result and the evaluation result to obtain at least one cluster diagnosis; and assembling the cluster diagnosis to obtain a report thereby identifying and diagnosing the presence of the disease or the condition in the subject. The evaluation result is
5 synonymous with AI result, AI processor result and classification result are used interchangeably.

An embodiment of the method further includes prior to classifying, obtaining at least one radiologic image or one data point of the subject. An embodiment of the method further includes prior to comparing, compiling the dataset cluster using a clustering tool selected
10 from: K-means clustering, Mean shift clustering, Density-Based Spatial Clustering, Expectation–Maximization (EM) Clustering, and Agglomerative Hierarchical Clustering. In an embodiment of the method, compiling further includes obtaining, processing, evaluating, and constructing a library of a plurality of identified and diagnosed dataset and corresponding medical reports selected from: radiology reports, laboratory reports, histology reports,
15 physical exam reports, and microbiology reports, with a plurality of known diseases or conditions. The phrase, “medical reports” includes any type of medical data. The words “radiologic” and “radiographic” shall have the identical meaning.

In an embodiment of the method, processing further includes classifying the plurality of identified and diagnosed dataset images to the body regions to obtain a plurality of
20 classified dataset images, and further includes orienting and cropping the plurality of classified dataset images to obtain a plurality of oriented, cropped and labeled dataset sub-images. In an embodiment of the method, evaluating further includes directing the plurality of oriented, cropped and labeled dataset sub-images and corresponding medical reports to at least one AI processor to obtain at least one diagnosed AI processor result. In an embodiment
25 of the method, directing further includes classifying the plurality of oriented, cropped and labeled dataset sub-images and corresponding medical reports with at least one variable selected from: species, breed, weight, sex, and location.

In an embodiment of the method, constructing the library of the plurality of identified and diagnosed dataset images further includes creating at least one cluster of the diagnosed
30 AI processor result to obtain at least one AI processor exemplar result and thereby compiling the dataset cluster. In some embodiments the AI processor exemplar result is an exemplar case, an exemplar result, an exemplar point, or an exemplar. These terms are synonymous and interchangeably used. An embodiment of the method further includes assigning at least one cluster diagnosis to the cluster of the diagnosed AI processor result. In an embodiment of

the method, assigning cluster diagnosis further includes adding reports within the cluster and/or additional information written by an evaluator. In an embodiment of the method, measuring further includes determining a distance between the cluster result and at least one selected from: the evaluation result, the dataset cluster, and a centroid of the cluster result.

5 An embodiment of the method further includes selecting a result from: a case within the cluster that has the nearest match, a result from another case in the cluster, and a centroid case. In an embodiment of the method selecting further includes adding result information of the cluster result by an evaluator to the report generated from the cluster. An embodiment of the method further includes editing the report by removing a portion of the report of the
10 cluster diagnosis which is less than a threshold of prevalence in a plurality of reports in the cluster. In an embodiment of the method, report is generated from words, partial sentences, sentences, and paragraphs that are deemed acceptable for use in report generation. The words in the report are obtained from the closest matching exemplar result case. The words that are acceptable for report generation are excluded if the words include at least one identifier
15 selected from: a subject name, date, reference to a prior study, or any other word that could generate a report that was not universally usable for all new cases that match closest to that exemplar result. This selection process is performed by Natural Language Processing (NLP) and language AI.

 In an embodiment of the method, the threshold of prevalence designated by an
20 evaluator. This threshold can be set to between 0.000001% and 99.999999%. In an embodiment of the method, rapidly processing the evaluation result by a diagnostic AI processor to obtain the report. In an embodiment of the method, the diagnostic AI processor processing the image to obtain the report within a very short time interval: less than about ten minutes, less than about 9 minutes, less than about 8 minutes, less than about 7 minutes, less
25 than about 6 minutes, less than about 5 minutes, less than about 4 minutes, less than about 3 minutes, less than about 2 minutes, or less than about 1 minute. In an embodiment of the method, the library of identified and diagnosed dataset images with known diseases and conditions are categorized to at least one of a plurality of animal species.

 An embodiment of the method further includes identifying the diagnosed AI
30 processor result with an identification tag. An embodiment of the method further includes selecting and adding the AI results from the image and/or a medical result of the subject to the database cluster, along with retaining the original and manipulated image.

 An aspect of the invention described herein provides a system for diagnosing a presence of a disease or a condition in an image and/or a medical result, of a subject, the

system including: a receiver to receive an image and/or the medical result of the subject; at least one processor to automatically run an image identification and processing algorithm to identify, crop, orient and label at least one body region in the image to obtain a sub-image; at least one artificial intelligence processor to evaluate the sub-image and/or the medical result and obtain an evaluation result; and at least one diagnostic artificial intelligence processor to automatically run a cluster algorithm to compare the evaluation result to obtain a cluster result, measure distance between cluster result and a previously created cluster result from a specific dataset defined by one or more variables, evaluation result to obtain cluster diagnosis, and assemble a report.

10 In an embodiment of the method, the diagnostic AI processor automatically rapidly processes the image, and/or the medical result to generate a report. In an embodiment of the method, the diagnostic AI processor processes the image and/or the medical result to obtain the report in: less than about ten minutes, less than about 9 minutes, less than about 8 minutes, less than about 7 minutes, less than about 6 minutes, less than about 5 minutes, less than about 4 minutes, less than about 3 minutes, less than about 2 minutes, or less than about 1 minute. An embodiment of the method further includes a device to display the generated report.

An aspect of the invention described herein provides a method for diagnosing a presence of a disease or a condition in at least one image of a subject, the method including: 20 classifying the image to at least one body region, labelling, cropping, and orientating the image to obtain at least one classified, labeled, cropped, and oriented sub-image; directing the sub-image to at least one artificial intelligence (AI) processor for processing and obtaining an evaluation result, and comparing the evaluation result to a database library having a plurality of evaluation results and a matched written templates or at least one dataset cluster to obtain 25 at least one cluster result; measuring the distance between the cluster result and the evaluation result to obtain at least one cluster diagnosis; and assembling the cluster diagnosis and the matched written templates to obtain a report and displaying the report thereby identifying and diagnosing the presence of the disease or the condition in the subject.

An embodiment of the method further includes after displaying, analyzing the report 30 and confirming the presence of the disease or the condition. An alternative embodiment of the method further includes editing the written templates. In an embodiment of the method, obtaining the report has a process time: less than about 5 minutes, less than about 2 minutes, or less than about 1 minute. In an embodiment of the method, obtaining the report has a

process time: less than about 10 minutes, less than about 7 minutes, or less than about 6 minutes.

In an embodiment of the method, processing the sub-image further includes training the AI processor for diagnosing the presence of the disease or the condition in the image of the subject. In an embodiment of the method, training the AI processor further includes the steps of: communicating a library of training images to the AI processor creating an AI model, storing the AI model on a database, testing the AI model using expected positive and negative data not used for training and comparing the actual test set results to expected test set results.

In an embodiment of the method, the library of training images includes positive control training images and negative control training images. In an embodiment of the method, the positive control training images have the disease or the condition of the training image. In an embodiment of the method, the negative control training images do not have the disease or the condition of the training image. In various embodiments of the method, the negative control training images may have diseases or conditions other than the disease or the condition of the training image. In an embodiment of the method, the library of training images further includes at least one of medical data, metadata, and auxiliary data.

An aspect of invention herein describes a novel system with several stages of analysis, including determining whether the received image includes a particular body region in a particular orientation (lateral view, etc.), cropping the image appropriately, and evaluating the cropped image by comparing the image to targeted AI models. In various embodiments, the newly received images are pre-processed to automatically identify and label one or more body regions and/or views that are represented in the image without user input or intervention. In some embodiments, the image is cropped automatically to generate one or more sub-images corresponding to the respective body regions/views that were identified. In some embodiments the image and/or sub-images are selectively processed to targeted AI processors which are configured to evaluate the identified body regions/views, excluding the remainder of the AI processors in the system.

In some embodiments, the radiologic image pre-processor 106 additionally or alternatively tags the entire image 102 to identify the body regions and/or views that were identified within the image 102, and then pass the entire image 102 to only those AI processors 104 that correspond to the applied tags. Accordingly, in such embodiments, the AI processors 104 are responsible for cropping the image 102 to focus on the pertinent regions

for further analysis using one or more trained machine learning models or otherwise. In some embodiments, in addition to tagging the image 102 as corresponding to particular body regions/views, the radiologic image pre-processor 106 additionally crops the image 102 to focus primarily on the regions of the image that actually represent portions of the animal and to remove as much of the black border around those regions as possible. In some
5 embodiments, performing such a cropping step facilitates further cropping and/or other processing by the AI processor(s) 104 that are subsequently deployed to evaluate particular body regions/views corresponding to the applied tags.

 The radiologic image pre-processor 106 is implemented in any of several ways. In
10 some embodiments, for example, the radiologic image pre-processor 106 employs one or more algorithms for identifying one or more features indicative of one or more particular body regions, and automatically cropping the image 102 to focus on those regions that include such features and/or on those regions that actually represent the animal. In some implementations, such algorithms are implemented, for example, using elements of the
15 OpenCV-Python library. A description of the Open Source Computer Vision (“OpenCV”) library, as well as documentation and tutorials concerning the same, is found using the uniform resource locator (URL) for OpenCV. The entire contents of the materials accessible via the URL are incorporated herein by reference. In some embodiments, the radiologic image pre-processor 106 additionally or alternatively employs image matching techniques to
20 compare the image 102 and/or one or more cropped sub-images 108 thereof against a repository of stored images that are known to represent particular views of specific body regions, and the image 102 and/or sub-images 108 are determined to represent the body region/view for which the strongest correlation is found with one or more of the stored images. In some embodiments, an AI processor trained to perform body region/view
25 identification additionally or alternatively is employed within the radiologic image pre-processor 106.

 In some embodiments, one or more of the AI processors described herein are implemented using the TensorFlow platform. A description of the TensorFlow platform, documentation and tutorials are found using TensorFlow website. The entire contents of the
30 materials accessible via the website are incorporated herein by reference. The TensorFlow platform and methods for building AI processors are fully described in Hope, Tom, et al. *Learning TensorFlow: A Guide to Building Deep Learning Systems*. O'Reilly., 2017 which is hereby incorporated by reference herein in its entirety.

In the example shown in FIG. 3, the pre-processing performed by the radiologic image pre-processor 106 included (1) an optional “general” auto-cropping step (reflected in the first five log entries delineated by the bracket 306) pursuant to which the image 302 was initially cropped to focus primarily on the regions of the image that represent portions of the animal and to remove as much of the black border around those regions as possible, (2) a
5 “classified” auto-cropping step (reflected in log entries six through nine within the bracket 306) pursuant to which an initial effort was made, e.g., using elements of the OpenCV-Python library, to identify particular body regions/views and crop the image 302 to focus on the same, and (3) an AI region labeling step or an “image matching” step (reflected in the
10 final three log entries delineated by the bracket 306) pursuant to which the image 302 and/or one or more cropped sub-images 304a-b thereof was compared to a repository of stored images that are known to represent particular views of specific body regions. As indicated by the corresponding time stamps, the general auto-cropping step was observed to be completed in two seconds, the classified auto-cropping step was observed to be completed in three
15 seconds, and the image matching step in nineteen seconds.

As shown by the log entries delineated by the bracket 308a in FIG. 3, the time taken by the lateral thorax AI processor 104a to determine whether the image 302 included a lateral view of an animal’s thorax was four seconds. Similarly, as indicated by the log entries delineated by the bracket 308b, the lateral abdomen AI processor 104c determined whether
20 the image 302 included a lateral view of the animal’s abdomen in four seconds.

Had the system instead needed to process the newly received image 302 with all of the possible AI processors 104a-l, rather than just the two AI processors corresponding to the body parts/views identified by the radiologic image pre-processor 106, the time by the AI processors would have been significantly longer and/or would have consumed significantly
25 more processing resources to complete the analysis. In a system including thirty different AI processors 104, for example, the processing simply to identify pertinent AI models for determining condition(s) of the imaged animal would have been at least one hundred and twenty seconds of processing time by the AI processors 104 (i.e., thirty AI processors at four seconds per processor), and likely much longer when multiple possible orientations of the
30 image are considered by each of the AI processors 104. By employing the radiologic image pre-processor 106, on the other hand, identification of the pertinent AI models was observed to be only eight seconds of processing time of the AI processors 104, plus twenty-four seconds of pre-processing time by the radiologic image pre-processor 106.

It is useful to process radiologic images using artificial intelligence (AI) processors, e.g., trained neural networks, to identify and label the image data and determine probabilities that the imaging data does or does not have certain medical conditions. Typically, separate AI processors are used to evaluate respective body regions (e.g., thorax, abdomen, shoulder, fore limbs, hind limbs, etc.) and/or particular orientations (e.g., ventral dorsal (VD) view, lateral view, etc.) of each such body region, with each such AI processor determining, for a respective body region and/or orientation, probabilities that particular conditions exist with respect to the body region in question. Each such AI processor may include a large number of trained models to evaluate respective conditions or organs within the imaged region. For example, with respect to a lateral view of an animal's thorax, an AI processor may employ different models to determine probabilities that the animal has certain conditions relating to the lungs, such as perihilar infiltrate, pneumonia, bronchitis, pulmonary nodules, etc.

The detection of a single disease condition, such as presence or absence of pneumonia or pneumothorax, is practiced in radiology AI at the present time. In contrast to single disease detection by current radiology AI, human radiologists analyze the radiographs in a whole-istic approach by evaluate the presence or absence of many conditions simultaneously. A limitation of the current AI process is the necessity to use a separate AI detector for each specific condition. However, a combination of conditions results in the diagnosis of a broader disease. For example, in some cases, one or more diagnostic results obtained from radiologic images are caused by several broader diseases. Determining the broader diseases that are present in the subject's radiograph requires use of supplemental diagnostic results in a process known as differential diagnosis. These supplemental diagnostic results are extracted from blood work, patient history, biopsies, or other tests and processes in addition to radiologic images. The current AI process is focused on single diagnostic results and is unable to identify broader diseases requiring differential diagnosis. A novel AI process which is able to combine multiple diagnostic results to diagnose broader diseases is described herein.

The AI process currently uses limited radiologic images which are directed to specific areas as is typical in radiologic images of human subjects. In contrast, veterinary radiology regularly includes multiple body regions within a single radiograph. A novel AI evaluation process to evaluate all body regions included in the study and providing a broader evaluation expected in veterinary radiology is described herein.

The current conventional workflow for AI reporting of a single disease process is illustrated in FIG.4. The conventional single condition reporting shown in FIG. 4 is

insufficient for differential diagnosis of radiographs. Further, using individualized rules for each combination of evaluation results is inefficient to create reports and is unable to meet reporting standards expected of veterinary radiologists. Even for a single disease process, a determination of the severity of a particular condition e.g. normal, minimal, mild, moderate, and severe results in an exponential number of AI models results templates. The process of creating and choosing a single report template from a collection AI model diagnostic results scales exponentially with the number of AI models. The number of AI models for a single disease process results in 57 different templates for five different severities as illustrated in FIG. 5A- FIG. 5F. Therefore, a manually created report for each combination of AI model diagnostic results does not scale well to a large number of AI models being interpreted together.

Automated system for AI analysis

Described herein is a novel system that analyzes images of a subject animal, the system including: a receiver to receive an image of the subject; at least one sub-image processor to automatically identify, crop, orient and label at least one body region in the image to obtain a sub-image; at least one artificial intelligence evaluation processor to evaluate the sub-image for presence of at least one condition; at least one synthesis processor to generate an overall result report from at least one sub-image evaluation and, optionally, non-image data; and a device to display the sub-images and an overall synthesized diagnostic result report.

The system provides a substantial advancement in veterinary diagnostic image analysis by (1) automating sub-image extraction using a sub-image processor, a task that typically occurs manually or with user assistance, and (2) by synthesizing a large collection of evaluation results and other non-image datapoints into a concise and cohesive overall report using a synthesis processor.

A case includes a collection of one or more images of a subject animal and may include non-image data points such as, but not limited to, age, sex, location, medical history, and other medical test results. In an embodiment of the system, each image is sent to multiple sub-image processors producing many sub-images of various views of multiple body regions. Each sub-image is processed by multiple evaluation processors, generating a multitude of evaluation results for many different conditions, findings, or other features spanning many body regions. A synthesis processor processes all or a subset of evaluation results and non-image data points to produce an overall synthesized diagnostic result report.

In an embodiment of the system, multiple synthesis processors produce multiple synthesized diagnostic result reports from differing subsets of evaluation results and non-image data points. These diagnostic reports are assembled with ancillary data together to create the final overall synthesized diagnostic result report.

5 In an embodiment of the system, each synthesis processor runs on a subset of sub-images and non-image data points corresponding to a body region, e.g., the thorax or abdomen. Each synthesized diagnostic report includes a body region, as is the typical practice in veterinary radiology. The overall synthesized diagnostic result report includes descriptive data of the subject, e.g., name, age, address, breed, and multiple sections corresponding to the
10 output of each synthesis processor, e.g., a thorax diagnostic result section and an abdomen diagnostic result section.

 In an embodiment of the system, the subject is selected from: a mammal, a reptile, a fish, an amphibian, a chordate, and a bird. The mammal is dog, cat, rodent, horse, sheep, cow, goat, camel, alpaca, water buffalo, elephant, and human. The subject is a pet, a farm animal, a
15 high value zoo animal, a wild animal, and a research animal.

 The images received by the system are images from a radiology exam such as X-ray(radiographs), magnetic resonance imaging (MRI), magnetic resonance angiography (MRA), computed tomography (CT), fluoroscopy, mammography, nuclear medicine, Positron emission tomography (PET), and ultrasound. In some embodiments, the images are
20 photographs.

 In some embodiments of the system, analyzing images of a subject generates and displays an overall synthesized result report in a very short time interval: less than about twenty minutes, less than about 10 minutes, less than about 5 minutes, less than about one minute, less than about 30 seconds, less than about 20 seconds, less than about 15 seconds,
25 less than about 10 seconds, or less than about 5 seconds.

Sub-image processor

 The sub-image processor orients, crops, and labels at least one body region in an image to obtain a sub-image automatically and rapidly. The sub-image processor orients the image by rotating the image into a standard orientation depending on the specific view. The
30 orientation is determined by veterinary radiograph standard hanging protocol. The sub-image processor crops the image by identifying a boundary in the image delineating one or more body regions and creating a sub-image containing image data within the identified boundary.

In some embodiments, the boundary is of a consistent aspect ratio. In alternative embodiments, the boundary is not of a consistent aspect ratio. The sub-image processor labels the sub-image by reporting boundary and/or location of each body region contained within the sub-image. Body regions are for example: thorax, abdomen, spine, forelimb, left
5 shoulder, head, neck, etc. In some embodiments the sub-image processor labels the sub-image according to veterinary radiologic standard body region labels.

The sub-image processor matches the image to a plurality of reference images in at least one of a plurality of libraries to orient, crop and label one or more sub-images. Each of the plurality of libraries include respective plurality of reference images specific or non-
10 specific to an animal species.

The sub-image processor extracts a signature of the image prior to orienting, cropping, and/or labeling the image, thereby allowing rapid matching of the image or sub-image to similar reference images. The sub-image processor processes the image to obtain the sub-image in a short time interval, for example, less than about twenty minutes, less than 10
15 minutes, less than about 5 minutes, less than about one minute, less than about 30 seconds, less than about 20 seconds, less than about 15 seconds, less than about 10 seconds, less than about 5 seconds, less than about 4 seconds, less than about 3 seconds, less than about 2 seconds, less than about 1 second, less than about 0.5 seconds and less than about 0.1 seconds.

20 Evaluation processor

The artificial intelligence evaluation processor assesses a sub-image for a presence or an absence of a condition, finding, or other feature. The evaluation processor reports the probability of presence of a condition, a finding, or a feature.

The evaluation processor diagnoses the presence or absence of a medical condition
25 from the sub-image. The evaluation processor assesses the sub-image for a non-medical feature, for example, proper positioning of the subject. The evaluation processor generates instructions for rectifying the positioning of the subject.

Typically, evaluation processor training includes negative control/normal and positive control/abnormal training sets with respect to a condition, finding, or other feature.
30 The positive control/abnormal training set typically includes cases in which presence of the condition, finding, or other feature has been assessed. The negative control/normal training set includes cases in which the absence of the condition, finding or other feature has been assessed and/or the cases are deemed completely normal. In some embodiments, the negative control/normal training set includes cases in which a presence of other conditions, findings,

or features distinct from the one of interest have been assessed. Therefore, the evaluation processor is robust.

The evaluation processor processes the sub-image to report the presence of the condition in: less than about twenty minutes, less than about 10 minutes, less than about 5 minutes, less than about one minute, less than about 30 seconds, less than about 20 seconds, less than about 15 seconds, less than about 10 seconds, less than about 5 seconds, less than about 4 seconds, less than about 3 seconds, less than about 2 seconds, less than about 1 second, less than about 0.5 seconds and less than about 0.1 seconds.

Synthesis processor

The synthesis processor receives at least one evaluation from an evaluation processor and generates a comprehensive result report. The synthesis processor may include non-image data points, for example species, breed, age, weight, location, sex, medical test history including blood, urine, and fecal tests, radiology reports, laboratory reports, histology reports, physical exam reports, microbiology reports, or other medical and non-medical tests or results. The subject's case exemplar result includes at least one image, associated evaluation processor results and a collection of zero or more up to date non-image data points.

In an embodiment of the method, the synthesis processor uses the case exemplar result to select a result stored in the database.

In an embodiment of the method may be pre-written word, keywords, partial sentence, complete sentence, partial paragraph, paragraph, and/or multiple paragraphs to output as an overall result report. The template is customized automatically based on case exemplar result elements to provide a customized overall result report.

The synthesis processor assigns the subject's case exemplar result to a cluster group. The cluster group contains other similar case exemplar results from a reference library of case exemplar results from other subjects. In some cases, the cluster group contains partial case exemplar results, e.g., a result report. The reference library includes case exemplar results with or without known diseases and conditions from at least one of a plurality of animal species. New case exemplar results are added to the reference library to improve the synthesis processor performance over time. The synthesis processor assigns coordinates representing the location of each case exemplar result within a cluster group.

A single overall result report is assigned to the entire cluster group and the overall result report is assigned to the subject by the synthesis processor. In some embodiments, several overall result reports are assigned to various case exemplar results within the cluster and/or various custom coordinates within the cluster, such as the cluster centroid, with no

associated case exemplar result. The coordinates of the subject's case exemplar result are used to calculate a distance to the nearest or non-nearest case exemplar result or custom coordinate that has an associated overall result report, which is then assigned to the subject.

The overall result report or reports are written by expert human evaluators. In an alternative embodiment, the overall result report or reports are generated from existing radiology reports. The content of existing radiology reports are linked to exemplar AI classifier results, modified by Natural Language Processing (NLP) or AI, to remove content that is not universally applicable, such as names, dates, references to prior studies, etc. to create suitable overall result reports. Statements contained within the overall result report are removed or edited if the statements do not meet a threshold of prevalence within the cluster.

In some embodiments, a personalized output report is created where an AI system is trained to best match an image evaluation and interpretation style of individual radiologists. In a personalized system, the nuances of an individual radiologist's image assessment and word, sentence, paragraph, grammar, report formatting choice is stored in a database and used to create a personalized and stylized report using standard AI classifiers, individualized AI result thresholding, individualized AI result weighting and radiologist specific AI language profiles.

The synthesis processor outputs the assigned overall result report for the subject, thereby identifying and diagnosing the presence of one or more findings, diseases and/or conditions in the subject. Cluster groups are established from a reference library of case exemplar results using a clustering tool selected from: K-means clustering, Mean shift clustering, Density-Based Spatial Clustering, Expectation–Maximization (EM) Clustering, and Agglomerative Hierarchical Clustering.

The synthesis processor processes the case exemplar result to generate an overall result report: less than about 20 minutes, less than about 10 minutes, less than about 9 minutes, less than about 8 minutes, less than about 7 minutes, less than about 6 minutes, less than about 5 minutes, less than about 4 minutes, less than about 3 minutes, less than about 2 minutes, less than about one minute, less than about 30 seconds, less than about 20 seconds, less than about 15 seconds, less than about 10 seconds, less than about 5 seconds, less than about 4 seconds, less than about 3 seconds, less than about 2 seconds, less than about 1 second, less than about 0.5 seconds and less than about 0.1 seconds.

Clustering is an AI technique for grouping unlabeled examples by similarities in features of each example. A process for clustering patient studies based on AI processor diagnostic results in addition to non-radiologic and/or non-AI diagnostic results is described

herein. The clustering process groups reports that share a similar diagnosis or output report, thereby facilitating a holistic detection of conditions or broader diseases in a scalable way.

A novel system and methods with multiple stages of analysis, combining multiple methods of AI predictive image analysis on the radiograph image and report library database
5 and newly received image evaluation to accurately diagnose and report radiology cases are described herein. In various embodiments, the novel system described herein automatically detects the view and regions or regions covered by each radiologic image.

In some embodiments, the system pre-processes the newly received radiologic image
102 to crop, rotate, flip, create sub-images and/or normalize the image exposure using a
10 radiologic image pre-processor 106 prior to AI evaluation. If more than one body region or view are identified, then the system further crops the image 102 to generate one or more sub-images 108a, 108b and 108c corresponding to the respective regions and view that were identified. In some embodiments, the system selectively processes and directs the image and/or sub-images to targeted AI processors configured to evaluate the identified
15 regions/view. The image 108a is directed only to AI processor 104a which is a lateral thorax AI processor. The image 108b is directed only to AI processor 104c which is a lateral abdomen AI processor. The image 108c is directed only to AI processor 104k which is lateral pelvis AI processor. The image is not directed to AI processors which are not targeted or to the remainder of the AI processors in the system. For example, thoracic image FIG. 7 is
20 directed to one or more AI processors for a disease listed in FIG. 6 such as heart failure, pneumonia, bronchitis, interstitial, diseased lung, hypoplastic trachea, cardiomegaly, pulmonary nodules, pleural effusion, gastritis, esophagitis, bronchiectasis, pulmonary hyperinflation, pulmonary vessel enlargement, thoracic lymphadenopathy, etc.

In some embodiments, the AI model processors are binary processors which provide a
25 binary result of normal or abnormal. In various embodiments, the AI model processors provide a normal or abnormal diagnosis with a determination of the severity of a particular condition e.g., severity of the condition can be classified as normal, minimal, mild, moderate, or severe.

In some embodiments, the newly received AI model processor results are displayed in
30 a user interface. See FIG. 9A-FIG. 9E. The mean AI model processor result for each model is collected from the individual image or sub-images evaluation results and displayed. See FIG. 9A. The user interface displays the individual image or sub-images and the AI model processor result for that image. See FIG. 9B-FIG. 9E. The AI analysis is completed in less than one minute, two minutes, or three minutes.

In some embodiments, one or more clusters are built by the system using AI processor diagnostic results from a library of known radiologic images and corresponding radiology reports database to develop a closest match case or an AI processor “exemplar result” for one or more AI processor results. An exemplar result includes at least one image, the collection of
5 associated evaluation processor results, and a collection of zero or more non-image data points such as age, sex, location, breed, medical test results, etc. The synthesis processor assigns coordinates representing the location of each case exemplar result within a cluster group. Therefore, if two cases have similar exemplar results, then the diagnosis is similar or largely identical and a single overall result report applies to the two cases. In some
10 embodiments, single exemplar result is assigned to an entire cluster and a subject case that is located in the cluster gets assigned the exemplar result. In some embodiments, multiple exemplar results are assigned to the cluster which are either tied to specific coordinates (e.g., the centroid) or specific dataset cases in the cluster. In some embodiments, exemplar results are written by a human or autogenerated from existing radiology reports tied to cases.

15 In some embodiments, the user specifies various parameters for creating the cluster from the library of known radiologic images and corresponding radiology reports database with a user interface of FIG. 11. The user assigns a name 1101 for the new cluster under “Code”. The user selects various parameters to create a cluster. The user chooses a start Case date 1102 and an end Case date 1103 to select the cases. The user chooses a start Case ID
20 (1104) and an end Case ID 1105 to select the cases. The user chooses a maximum number of cases 1106 that are to be included in the cluster. The user chooses species 1107 such as dog, cat, dog or cat, human, avian pet, farm animal, etc. for the cases to be included in the cluster. The user selects a specific diagnostic modality 1108 such as X-ray, CT, MRI, blood analysis, urinalysis, etc. to be included in creating the cluster.

25 In various embodiments, the user specifies separating the evaluation results into a specific number of clusters. The number of clusters range from a minimum of one cluster to a maximum of clusters limited only by total number of cases entered into the cluster. The system builds one or more clusters using non-radiologic and/or non-AI diagnostic results, such as blood work, patient history, or other tests or processes, in addition to AI processor
30 diagnostic results. The clusters are listed in a numerical form in a comma separated value (CSV) file format as shown in FIG. 12. The CSV file lists the case IDs 1201 of the cases in the cluster. The mean evaluation results 1202 for each binary model for the specific case ID are listed in the CSV file. The cluster label or cluster location 1203 which includes the specific case based on the collection of evaluation results are listed in the CSV file. The CSV

files lists the cluster coordinates. The case ID 1204 of the centroid or center of the specific cluster is listed in the CSV file.

In various embodiments, the cluster is represented by a clustering graph. See FIG. 13. The clustering graph is created by dividing the mean evaluation results into multiple different clusters depending on user defined parameters 1102-1108. The various different clusters are represented by a collection of dots plotted on the graph. The clustering graph of FIG. 13 shows 180 clusters of various sizes.

In some embodiments, a user interface shows AI cluster models generated based on user defined parameters 1102-1108. See FIG. 14. A user interface shows screening evaluation configuration in which a user assigns a specific "cluster model" 1502 to a specific "screening evaluation configuration" name 1501. The status 1503 of the screening evaluation configuration provides additional information about the configuration such as whether the configuration is in live, testing, or draft mode. The live mode is for production and the testing or draft mode is for development.

In some embodiments, a user interface describes the details for a specific cluster model 1601 Thorax 97. See FIG. 16A. In some embodiments, the user interface lists the AI evaluation classifier types 1602 included in the cluster. The user interface displays additional parameters used for building the cluster such as the species 1603 specific for the cluster model, the maximum number of cases 1604 with evaluation results, or the start and end dates 1605 for cases used to create the cluster. The user interface provides a link 1606 to the comma separated value (CSV) file showing the cluster in numerical table format. The user interface lists sub-clusters 1608 created from the parameters 1602-1605. The user interface displays the total number of sub-clusters 1609 created for the cluster group. The user interface provides a centroid case ID 1610 for each sub-cluster. The log for building the cluster is provided in the user interface. See FIG. 16B.

In various embodiments, the system utilizes one or more AI processors to evaluate newly received undiagnosed images and obtain newly received evaluation results. The system compares the newly received evaluation results to one or more clusters obtained from the library of known radiologic images and corresponding radiology reports database.

The user imports the newly received AI processor results into an AI Eval tester. See FIG. 17A. The user specifies the screening evaluation type 1702 and the corresponding cluster model 1703.

The system compares the non-radiologic and/or non-AI diagnostic results in addition to the newly received evaluation results to one or more clusters obtained from the library of

known radiologic images and corresponding radiology reports database in addition to other available. The system measures the distance between the location of newly received AI processor results and the cluster results and utilizes one or more cluster results to create a radiologist report. In some embodiments, the system chooses to utilize the entire radiologist report or a portion of a radiologist report from the known cluster results depending on the location of the newly received AI processor results relative to the known cluster results. In various embodiments, the system chooses to utilize the entire radiologist report or a portion of the radiologist report from other results in the same cluster. In some embodiments, the system chooses to utilize the entire radiologist report or a portion of the radiologist report from the centroid of the cluster result.

A user interface displays the result of the AI Eval tester. See FIG. 18A. In various embodiments, the diagnosis and conclusory findings 1801 from a radiologist report that are closest to the evaluation results based on previously created cluster results are displayed. In some embodiments, evaluation findings 1802 are selected from radiologist reports in the cluster of the evaluation results and filtered based on the prevalence of a specific sentence in the findings section of the specific cluster. In some embodiments, recommendations 1803 from the radiologist report in the cluster are selected based on prevalence of each sentence or a similar sentence in the recommendations section of this cluster. The user interface displays the radiologist report of the cluster 1804 and the radiologist report of the centroid of the cluster 1805. A user interface allows the user to edit the report by editing the findings section 1809, the conclusion section 1810 or the recommendation section 1811 by adding or removing specific sentences. See FIG. 18D and FIG. 18E. A radiologist report for the closest match database case is used to generate the radiology report for the newly received radiologic image. The sentences in a radiology report based on the specific cluster results are ranked and listed according to the rank and prevalence. See FIG. 18A and FIG. 18B.

In various embodiments, the system utilizes one or more AI processors to evaluate newly received undiagnosed images and obtain newly received evaluation results. The system compares the newly received evaluation results to one or more clusters obtained from the library of known radiologic images and corresponding radiology reports database.

The system compares the non-radiologic and/or non-AI diagnostic results in addition to the newly received evaluation results to one or more clusters obtained from the library of known radiologic images and corresponding radiology reports database in addition to other available. The system measures the distance between the location of newly received AI processor results and the cluster results and utilizes one or more cluster results to create a

radiologist report. In some embodiments, the system chooses to utilize the entire radiologist report or a portion of a radiologist report from the known cluster results depending on the location of the newly received AI processor results relative to the known cluster results. In various embodiments, the system chooses to utilize the entire radiologist report or a portion of the radiologist report from other results in the same cluster. In some embodiments, the system chooses to utilize the entire radiologist report or a portion of the radiologist report from the centroid of the cluster result.

In some embodiments, one or more of the AI processors described herein are implemented using the TensorFlow platform. A description of the TensorFlow platform, as well as documentation and tutorials concerning the same, is found on the TensorFlow website. The entire contents of the materials accessible in the TensorFlow website are incorporated herein by reference in its entirety.

In some embodiments, one or more of the clustering models described herein are implemented using the Plotly platform. A description of the Plotly platform, as well as documentation and tutorials concerning the same, are found on scikit learn website. The entire contents of the materials accessible on the scikit website are incorporated herein by reference in its entirety. The methods for developing AI processors and clustering models using TensorFlow platform and Scikit learn are fully described in the following references: Géron Aurélien. *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly, 2019; Hope, Tom, et al. *Learning TensorFlow: A Guide to Building Deep Learning Systems*. O'Reilly., 2017; and Sievert, Carson. *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*. CRC Press, 2020. Each of these references are hereby incorporated by reference herein in its entirety.

Had a radiologist attempted to evaluate each model and create rules based on separate AI processor results being found together, the creation of the rules and reports would be time prohibitive. Additionally, the addition of a single additional AI processor model into this scenario becomes exponentially more difficult as the number of already incorporated AI processor models increases. By employing the novel workflow of AI processor result clustering or "exemplar result" comparison between a new image and a known dataset to create the radiologist report, the issue of manual report building when multiple AI processor results are found is resolved. Report building manually via individual AI processor results and rule creation previously took months and with the novel workflow, only takes minimal time.

In some embodiments of the system, the components used for the AI evaluation are as described in FIG. 21A. In various embodiments of the system, the components used for image match AI processing are as described in FIG. 21B.

In various embodiments of the system, the server architecture for AI radiograph analysis includes pre-processing the radiologic images, analyzing the images using AI diagnostic processors and preparing reports based on clustering results. See FIG. 22B. In some embodiments, various servers are used including NGINX load balancing server 2202, V2 cloud platform 2203, database Microsoft SQL server 2207, and the datastore server 2208.

In various embodiments of the system, a user flags a case for training the AI system. In some embodiments of the system, the user flags cases if the radiology report requires editing because the radiology report is inaccurate, or if the report is inadequate or if the case has a novel diagnosis and hence the radiology report requires new language for diagnosis.

The series of schematics of AI autocropping and evaluation workflow are illustrated in FIG. 23A-FIG. 23F. The user accesses the V2 end user application 2301 to upload the image (in an image format such as DICOM, JPEG, JPG, PNG, etc.) to be analyzed by the system. In some embodiments the image is uploaded 2305 in the VetImages application directly. V2 processes 2302 the image, saves it to Datastore and requests 2303 VetImages to further process the image. VetImages receives the request from V2 and begins 2304 asynchronous processing. VetImages accesses 2307 the image from Datastore and requests 2308 VetConsole to preprocess the image. VetConsole uses OpenCV 2309 to improve the quality of the image and auto-crop 2310 the image. The tasks after accessing the image from Datastore are performed by the Sub-image processor.

The VetConsole sends the auto-cropped image with improved quality to VetImages which analyzes the image and requests VetConsole to classify 2311 thorax, abdomen and pelvis in the image. VetConsole classifies 2312 thorax, abdomen and pelvis in the image and sends coordinates to VetImages. The VetImages sends the image and the coordinates to ImageMatch validation 2313. The ImageMatch validation matches the image and the coordinates with correctly classified images in its database and sends 2314 the matched image distances and paths to VetImages. The VetImages application receives data for matched images and uses the database information to confirm 2315 the body region. The next task is to determine image orientation. The image is rotated and flipped 2317. After each rotation and flip the image is sent 2318 to ImageMatch orientation application to be compared to matched images and to measure distance and image paths between matched images and the newly received image. The ImageMatch orientation application sends results

2319 with distance between the newly received image and the matched images and image paths. The orientation of the newly received image that has the least distance from the matched image is selected 2320 by the VetImages application. The process of checking each orientation and each flip is repeated till the image is rotated 360 degrees and flipped at
5 appropriate angles. In some embodiments, the image with selected orientation is sent to VetAI to detect thorax 2321 and abdomen 2323 and obtain coordinates for cropping the image to obtain sub-images with thorax 2322 and abdomen 2324. The process of obtaining coordinates is coordinated with TensorFlow.

The VetImages application obtains the coordinates from ImageMatch validation and
10 crops 2325 the images according to the coordinates to obtain sub-images. The sub-images are sent 2326 to ImageMatch Validation application for matching. The database images are matched 2327 to the sub-images and the distance between the matched database images and sub-images and the image paths are sent to the VetImages application. The VetImages application receives 2328 the distance and image path data and confirms the body region
15 using the data received from the matched images. The VetImages application analyzes each sub-image to check if each sub-image is valid. If the sub-image(s) is not valid 2331 then the general cropped image from VetConsole application is saved in the database or datastore 2332. If the sub-image(s) is valid 2330 then the sub-image is saved in the database or datastore 2332. The image saved in the database or the datastore is the cropped image used
20 for further processing or analysis. The VetImages application saves the data obtained for the sub-image or the general cropped image from VetConsole in the database 2332.

The following tasks are performed by the evaluation processor. The VetImages application sends the cropped image for positioning evaluation 2333 to the VetAI application. The data received 2334 from the VetAI application by the VetImages application is saved in
25 database 2335 and a signal is sent to V2 application to send an email 2336 to the clinic. The VetImages application accesses 2337 live AI models from database. The cropped image is sent 2339 to appropriate AI models in the VetAI application based on the body region of the cropped image. The appropriate AI models are predetermined for each body region. The VetAI application sends the AI evaluation label and the machine learning (ML) AI evaluation
30 result 2340 to the VetImages application which saves 2341 these data in the database for the cropped image. The VetImages application calculates 2342 the label and the probability for image based on the AI evaluation results for the cropped image. The process of sending 2339 cropped images to obtain the AI evaluation result is reiterated 2342 until the predetermined AI models are processed 2338.

The VetImages application analyzes 2344 if each image from the case is processed to obtain AI evaluation result. If all images from the case are not processed, then VetImages returns to process the next image in the case. If all images from the case are processed, then VetImages calculates 2345 the label and the probability of the case as a whole based on the label and the probability of each cropped image. The VetImages application then changes 2346 its status to live and screening evaluation types from the database. Upon changing the status of VetImages application to live, the tasks are performed by the synthesis processor. The VetImages application assesses 2347 whether all screening evaluations are completed. If all screening evaluations are not completed, then the VetImages application assesses 2348 if the screening evaluation has to be completed by clustering. If the screening evaluation has to be completed by clustering, then the AI evaluation results for the processed images is sent 2349 to the VetAI application and the Best match cluster results are sent 2350 to VetImages application which generates and saves the screening results 2351 based on the best match cluster results in the database. If VetImages application determines that the screening evaluation is not to be performed with clustering, then the finding rules are accessed 2352 and the AI evaluation results are processed based on the finding rules to obtain and save 2353 screening results in the database. The process of obtaining screening results and saving in the database is reiterated until screening evaluations for all images in the case are completed and a complete result report is obtained 2354.

The VetImages application assesses 2355 if the species of the subject has been identified and saved in the database. If the species has not been identified, then the VetAI application evaluates 2357 the species of the subject and sends species evaluation results to VetImages application. The tasks for species evaluation 2356-2357 are performed by the Evaluation processor. In some embodiments, the VetImages application assesses 2358 if the species is canine. If the species is positively identified, for example, as canine then the case is flagged 2359 and the evaluation is attached to the result report. The VetImages application notifies 2360 V2 that the evaluation of the case is completed. V2 application assesses 2361 if the case is flagged. The result report is saved 2362 in the case documents if the report is flagged, and the result report is emailed 2363 to the client. If the report is not flagged then the result report is emailed 2363 to the client without saving the report to case documents.

Clustering

Clustering is a type of semi-supervised learning method. A semi-supervised learning method is a method in which references from datasets consisting of input data without labeled responses are drawn. Generally, clustering is used as a process to find meaningful structure,

explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is a task of dividing the population or data points into a number of groups such that the data points in the same groups are similar to other data points in the same group and dissimilar to the data points in other groups. Therefore, clustering is a method of collecting objects into groups on the basis of similarity and dissimilarity between them.

Clustering is an important process as it determines the intrinsic grouping among the data. There are no criteria for a good clustering as clustering depends on the user to choose a criterion that is useful to meet the purpose of the user. For example, clusters are based on finding representatives for homogeneous groups (data reduction), finding “natural clusters” and describing their unknown properties (“natural” data types), finding useful and suitable groupings (“useful” data classes) or finding unusual data objects (outlier detection). This algorithm makes assumptions which constitute the similarity of points and each assumption makes different and equally valid clusters.

Clustering Methods:

There are various methods for clustering which are as follows:

Density-based methods: These methods consider the cluster as a dense region having some similarity and differs from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters. Examples of density-based methods are Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Ordering Points to Identify Clustering Structure (OPTICS), etc.

Hierarchical based methods: The clusters formed by this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed clusters. The hierarchical based methods are divided into two categories: Agglomerative (bottom up approach) and Divisive (top down approach). Examples of hierarchical based methods include: Clustering Using Representatives (CURE), Balanced Iterative Reducing Clustering, Hierarchies (BIRCH), etc.

Partitioning methods: The partitioning methods divide the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function. Examples of partitioning methods include: K-means, Clustering Large Applications based upon Randomized Search (CLARANS), etc.

Grid-based methods: In grid-based method the data space is formulated into a finite number of cells that form a grid-like structure. All the clustering operation performed on these grids are fast and independent of the number of data objects. Examples of grid-based

methods are: Statistical Information Grid (STING), wave cluster, CLustering In Quest (CLIQUE), etc.

K-means clustering

K-means clustering is one of the unsupervised machine learning algorithms.

- 5 Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known or labelled outcomes. The objective of K-means is to simply group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.

- 10 A cluster refers to a collection of data points aggregated together because of certain similarities. A target number k refers to the number of centroids that a user would require in a dataset. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares. The K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

- 15 The 'means' in the K-means refers to averaging of the data which is finding the centroid. To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids. The algorithm halts creating and optimizing clusters when the centroids have
20 stabilized and there are no changes in their values because the clustering has been successful, or the defined number of iterations has been achieved.

- The method for K-mean clustering follows a simple method to classify a given data set through a certain number of clusters (assume k clusters) fixed *a priori*. The k centers or centroids one for each cluster are defined. The next step is to take each point belonging to a
25 given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is performed. The next step is to re-calculate k new centroids as barycenter of the clusters resulting from the previous step. Upon calculating the k new centroids, a new binding is performed between the same data set points and the nearest new center thereby generating a loop. As a result of this loop the k centers change their
30 location step by step until the k centers stop changing their location. The K-means cluster algorithm aims at minimizing an objective function know as squared error function which is calculated by the formula:

$$J(V) = \sum_{i=1}^c \sum_{j=1}^{c_i} (\|x_i - v_j\|)^2$$

in which,

' $\|x_i - v_j\|$ ' is the Euclidean distance between x_i and v_j .

' c_i ' is the number of data points in i^{th} cluster.

5 'c' is the number of cluster centers.

Algorithmic steps for K-means clustering

The algorithm for K-means clustering is as follows:

In K-means clustering 'c' cluster centers are randomly selected, the distance between each data point and cluster centers is calculated, the data point is assigned to the closest
10 cluster center, the new cluster centers are recalculated using the formula

$$v_i = (1/c_i) \sum_{j=1}^{c_i} x_i$$

where, ' c_i ' is the number of data points in i^{th} cluster,

X is $\{x_1, x_2, x_3, \dots, x_n\}$ the set of data points, and

V is $\{v_1, v_2, \dots, v_c\}$ the set of centers.

15 The distance between each data point and newly obtained cluster centers is measured and if a data point is reassigned then the process continues till no data point is reassigned.

AI application process flow

In some embodiments, Digital Imaging and Communications in Medicine (DICOM) images are submitted via LION and transmitted over Hypertext Transfer Protocol Secure
20 (HTTPS) by DICOM ToolKit library (Offis.de DCMTEK library) to a V2 platform. The DICOM images are temporarily stored in the V2 platform. In some embodiments a DICOM record is created with limited information and the status is set to zero. In various embodiments, once the DICOM images are available in temporary storage a V2 PHP/Laravel application begins processing the DICOM images through a Cron job.

25 In some embodiments, Cron job (1) monitors V2 for new DICOM images, obtains the DICOM images from temporary storage, extracts tags, extracts frames (single sub-image or multi sub-images), saves the images and tags in a data store and sets the processing status in a database. In some embodiments, Cron job (1) converts and compresses the DICOM images into a lossless JPG format using the Offis.de library DCMTEK and sets the processing status to
30 one. In some embodiments, the Cron job (1) automatically runs every few minutes such as

every five minutes, every four minutes, every three minutes, every two minutes or every minute. In some embodiments, the Cron job (1) saves DICOM image metadata to a table called DICOM in a Microsoft SQL server and extracts the images/frames and stores the images/frames in a directory for the image manager. In various embodiments, records are
5 created during processing which contain additional information about the image and the case ID associated with the image. The records contain additional data such as physical exam findings for the subject, study of multiple visits for the subject, series of images obtained during each exam, and hierarchy of the images within the case.

In various embodiments, the DICOM image and metadata are processed by a Vet
10 Images application written in PHP Laravel Framework. V2 makes a REST service request to VetImages to process each image asynchronously. In some embodiments, VetImages responds to V2 immediately to confirm that the request has been received and that the processes for cropping and evaluation of the images will continue in background. Because the images are processed in parallel the overall process is performed at high speed.

15 In various embodiments, VetImages passes or transfers the image to a module called VetConsole, which is written in Python and uses Computer Vision technology OpenCV to preprocess the image. VetConsole identifies body regions in the image such as thorax, abdomen, pelvis, as a reserve in case the AI Cropping server is unable to classify body region in the image. VetImages rotates and flips the image until a correct orientation is achieved. In
20 some embodiments, VetImages uses image match servers to validate the different angles and projections of the image. In various embodiments, the image match servers are written in Python and Elastic Search to identify image matches. In some embodiments, the image database for the image match servers are carefully selected to return results only if the image is accepted to be in correct orientation and projection.

25 In various embodiments, upon determining the orientation of the image VetImages sends a REST API request to Keras/TensorFlow server to classify and determine the region of interest of the body regions in the image. The VetImages REST API request is validated using Image Match servers to confirm that the returned regions of the image are classified into body regions such as thorax, abdomen, pelvis, stifles, etc. In some embodiments, if the
30 evaluation result for cropping is invalid, VetConsole cropping result is validated and utilized.

In various embodiments, the AI Evaluation process to generate the AI report is launched if VetImages determines that the image contains a classified and validated body region. In alternative embodiments, if VetImages determines that the image does not contain

a classified and validated body region, the cropping image process ends without results and without generation of a report.

In various embodiments, VetImages sends a REST service call to a Keras/TensorFlow with the classified cropped image to AI Evaluation models for diseases hosted on the
5 TensorFlow application servers written in Python/Django. VetImages saves the results of the AI evaluation models for final evaluation and for report generation.

VetImages also directs the thorax cropped images to TensorFlow server to determine if the image is well positioned with respect to the parameters set by the user. VetImages sends the results of the AI evaluation models to V2 Platform to notify the Clinic the results of
10 the positioning evaluation per image.

In some embodiments, VetImages waits while images are being processed in parallel until all images of a case are cropped and evaluated by the TensorFlow server. In some embodiments, upon completing the evaluation of all images of a case, VetImages process all results of the case using rules defined by a specialist to determine the content of the report in
15 a more human readable way. In an alternate embodiment, VetImages uses a Clustering model created by the user to determine the content of the AI report. In some embodiments, the AI report is assembled using previous radiologist reports which are used to build the Cluster Model in VetImages. In some embodiments, clustering is used to classify the case/image using the prediction results from other diagnostic models using scikit-learn.

In some embodiments, upon determining the content of the AI Report using Specialist
20 Rules or Cluster Models, VetImages checks the species of the case. In some embodiments, only if the species of the case is determined by VetImages to be canine, the report is generated and sent to V2 clinic.

In some embodiments, VetImages sends a request to V2 Platform to notify the Clinic
25 that the new AI report has been sent to the Clinic. In some embodiments, the V2 Platform validates the clinic admin user's or users' licenses. In various embodiments, V2 attaches a copy of the report in or to the Case Documents so that the report is accessible from V1 platform if the clinic has a valid license. In some embodiments, V2 sends an email notification to clinic emails, containing a link or links so that the email receiver can
30 conveniently and immediately open the generated reports.

Additive AI Models

The standard of practice for AI model retraining is to improve the current model performance by adding additional data, retraining, testing, and then replacing the current model with an updated AI model (Fig. 24). Under the current standard, good practice is to

leave the old model for a specific window or period or until the model has served a particular number of requests. A new model is provided the same data and the results are analyzed and compared to the old model. Therefore, the model which is performing better is identified. If the performance of the new model is satisfactory, the new model is deployed. The current
5 standard is a typical example of A/B testing, which ensures that the model is validated on upstream data. Another best practice is to automate the procedure of deployment of models after retraining. The machine learning models are deployed to the production environment using Kubernetes (K8s) which is an open-source system for automating deployment, scaling, and management of containerized applications.

10 In the current approach, it is assumed that the newly trained classifier performance is improved by reducing unwanted “noise” or false positive data. The assumed labeling of data detected as unwanted “noise” with the current classifier and removing this “noise” with the retrained classifier results from an incorrect assumption that the “noise” is not valuable information in overall system performance.

15 The fact, in contrast to the above incorrect assumption, is that the “noise” being detected by the current classifier is useful to distinguish data points with similar but not identical characteristics. The AI classifier results from the current classifier and new classifier upon combined analysis can recognize and categorize data better compared to either AI classifier alone.

20 The “noise” being detected by the current model is useful to distinguish data points with similar but not identical characteristics. Therefore, by replacing the current AI model, valuable data may be lost. The concept of learning in the human brain is based upon a foundation of knowledge and iterative learning steps. Along the path of iterative learning, prior knowledge is not lost or forgotten, rather prior knowledge is built upon and adjusted as
25 needed, adding to the overall bank of one’s knowledge and capabilities. Accordingly, the concept of building upon prior knowledge is applied to daisy chaining iterative or related model deployment.

Embodiments of methods described herein allow for continual training by iterative or related model deployment or daisy chaining iterative or related models rather than the current
30 standard of model retrain and replace (Fig. 25). These methods improve AI reporting performance, internal AI auditing with the system thereby progressing towards artificial general intelligence (AGI), if not creating a working artificial general intelligence system (AGI).

For example, an AI classifier built to detect a person with blond hair can be very broad and detect any person with any part of blond hair. A retrained AI classifier can be more specific and detect only people with 80% or more blond hair. Under the current best practice in the industry, the AI classifier is removed from service and replaced with the retrained classifier.

The scenario 3 in Table 1 herein demonstrates that an error result is easily detectable if the results from both AI classifiers are evaluated compared to a single, newly trained AI classifier result. In scenario 3, the old classifier is more sensitive but less specific compared to the new classifier. Therefore, the cases that the old classifier evaluates as positive for blond hair are also evaluated as positive by the new classifier. However, a negative result from the old or the sensitive classifier and a positive result for the new or the specific classifier is an erroneous result and is immediately detected and apparent in a multi-classifier system. Therefore, the methods provided herein include rules that can be implemented within the system such that the system would continue to operate different versions of the same classifier to detect outlier results more rapidly.

Table 1: Potential scenarios in a single classifier vs. dual classifier

Case	Scenarios	New Classifier Only	Current / New Classifier together
1.	Person with 90% blond hair	Positive	Positive / Positive
2.	Person with 70% blond hair	Negative	Positive / Negative
3.	Person with 70% blond hair and another unexpected variable such as strong sunlight or hair clip	Positive	Negative / Positive

The methods described herein show implementing an additive of derivative models into a production system. In the additive model approach, the user understands that a first model in production has some limitation and is retraining the model to improve upon a deficiency. As subsequent AI models are built, the original or first iteration model is maintained and the results from the first and the subsequent AI models are included in the AI cluster analysis as described in U.S. patent application publication number U.S. 2021/0202092A1 which is incorporated by reference herein in its entirety. The AI data results from old and new models are added into a single row of results and an AI fingerprint is

created for the item being evaluated. The multiple AI fingerprints are then clustered together to group similar results together. This approach of adding in related or derivative AI classifiers is defined as daisy chaining related or iterative, or unrelated or non-iterative AI classifiers along with accompanying words, phrases, sentences and paragraphs and non-AI derived data (Figs. 26 and 27).

FIG. 28 shows visual examples of both derivative (same letter and subscript number) and non-derivative (same letter but different subscript letter) data together. The database holds all the training data with the relationships. This system allows for data inputs (in this case I, T and OI); single or clustered, AI or non-AI data, derivative or non-derivative data, to obtain an exemplar result from the database. In FIG 28, the approach is subdivided into components A, B and C, shown by bounded boxes.

The component in box A shows a visual example showing how clustered data is both grouped and related to train the system (i.e., adding information to the database) with derivative or non-derivative results (training data) AND as the output step where a clustered derivative grouping of Image (I) or Text (T) is input into the system and the corresponding result is brought back from the database as the best exemplar result. In this case, derivative data is being shown as the input in images (I) and text (T).

The component in box B shows a visual example showing a clustered input, derivative data (Input data) and the corresponding result (Output pulled from database) in a stacked visual arrangement.

The component in box C shows a visual example showing a clustered input of derivative data (Input) and the corresponding result (Output) in a horizontal visual arrangement. Having only derivative classifiers are not a prerequisite of a clustering input or output and in this example OI in the output has subscripted letters OI_A and OI_B indicating non-derivative data. The different sized boxes denote the different weighting of the results. Additionally, the data represented by I, T and OI should be thought of as building blocks that when grouped together form instructions for building an output. That output can be a pixel, collection of pixels, image or collection of images grouped together, a color or certain arrangement of colors, a word or several words, a partial sentence or a complete sentence, several sentences together creating a partial paragraph, a full paragraph or several paragraphs or a full template, article, essay or other text based output, a type of measurement, a number, a waveform, an equation, a recipe, an atomic structure.

Like DNA primers, AI and non-AI data, both derivative and non-derivative data linked together into unique fingerprints by clustering, are groupings of results that serve as instructions for a specific output when applied against a complementary database and specific user defined presets. The novel concept here is to utilize AI and non-AI derived input data in a structured, linear format to serve as one or more “primer” instructions for generating corollary results from a vast database of previously stored “primer” results and having a computerized system built around the data fingerprint to act as the transcription machinery to create an output. This fingerprint or clustering “primer” approach allows for an unlimited possibility of output results thereby creating an extremely robust system (FIG. 28).

By daisy chaining related or iterative classifiers into the AI evaluation, the results undergo an additional sanity check (FIGS. 29 A and FIG. 29 B). The methods described herein show implementing an additive of derivative models into a production system. In the additive model approach, the user understands that a first model in production has some limitation and is retraining the model to improve upon a deficiency. As subsequent AI models are built, the original or first iteration model is maintained and the results from the first and the subsequent AI models are included in the AI cluster analysis as described in U.S. patent application publication number U.S. 2021/0202092A1 which is incorporated by reference herein in its entirety. The AI data results from old and new models are added into a single row of results and an AI fingerprint is created for the item being evaluated. The multiple AI fingerprints are then clustered together to group similar results together. This approach of adding in related or derivative AI classifiers is defined as daisy chaining related or iterative AI classifiers.

By daisy chaining related or iterative classifiers into the AI evaluation, the results undergo an additional sanity check. The sanity check allows the original classifier to cross reference results from the subsequent classifiers. Further the sanity check also identifies any newly introduced AI evaluation by the more specific classifier if the original or broader classifier has a negative result for a case, and the specific or newly trained classifier has a positive result.

In the examples of FIGS. 29A and 29B, the general lung classifier (GLC) is used with a derivative classifier (GLCn), or GLC2 for this specific example, as confirmation of each other. GLC2 is a more specific bronchial pattern classifier. Results for GLC are on the left and GLC2 are on the right. The general and specific classifier results are used as an internal check of each other for a finding. In these examples, the classifier results are expected to track with each other because example 8a has no abnormal finding and example 8b has an

abnormal finding that is picked up by both the GLC and its derivative classifier GLC2.

Depending on the specific image finding and classifiers used, the acceptable combinations of GLC and GLC2 are both negative for a finding (<0.5), both positive for a finding (>0.5) or GLC >0.5 and GLC2 <0.5 . Any result with GLC <0.5 and GLC2 >0.5 is considered

5 abnormal and a system error.

Methods for machine learning, clustering, and programming are fully described in the following references: Shaw, Zed. *Learn Python the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code*. Addison-Wesley, 2017; Ramalho, Luciano. *Fluent Python*. O'Reilly, 2016; Atienza, Rowel. *Advanced Deep Learning with*
10 *TensorFlow 2 and Keras: Apply DL, GANs, VAEs, Deep RL, Unsupervised Learning, Object Detection and Segmentation, and More*. Packt, 2020; Vincent, William S. *Django for Professionals: Production Websites with Python & Django*. Still River Press, 2020; Bradski, Gary R., and Adrian Kaehler. *Learning OpenCV*. O'Reilly, 2011; Battiti, Roberto, and Mauro Brunato. *The LION Way: Machine Learning plus Intelligent Optimization, Version 2.0, April*
15 *2015*. LIONlab Trento University, 2015; Piatnykh, Oleg S. *Digital Imaging and Communications in Medicine (DICOM) a Practical Introduction and Survival Guide*. Springer Berlin Heidelberg, 2012; Busuioc, Alexandru. *The PHP Workshop: a New, Interactive Approach to Learning PHP*. Packt Publishing, Limited, 2019; Stauffer, Matt. *Laravel - Up and Running: a Framework for Building Modern PHP Apps*. O'Reilly Media,
20 Incorporated, 2019; Kassambara, Alboukadel. *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA, 2017; and Wu, Junjie. *Advances in K-Means Clustering: A Data Mining Thinking*. Springer, 2012. Each of these references are hereby incorporated by reference herein in its entirety.

It is understood that any feature described in relation to any one of the embodiments
25 provided herein may be used alone, or in combination with other features described, and may also be used in combination with one or more features of any other of the embodiments, or any combination of any other of the embodiments. Furthermore, equivalents and modifications not described above may also be employed without departing from the scope of the invention, which is defined in the accompanying claims.

30 The invention now having been fully described, it is further exemplified by the following claims. Those skilled in the art will recognize or be able to ascertain using no more than routine experimentation, numerous equivalents to the specific methods described herein. Such equivalents are within the scope of the present invention and claims. The contents of all

references including issued patents and published patent applications cited in this application are hereby incorporated by reference in their entirety.

What is claimed is:

1. A method for obtaining an additive AI result from a digital file, the method comprising:
processing the digital file by at least one first artificial intelligence (AI) classifier and
5 at least one second AI classifier thereby obtaining a first evaluation result and at least one second evaluation result respectively;
directing the first evaluation result and the at least one second evaluation result to at least one synthesis processor;
creating a cluster of AI classifier and non-AI classifier results in a database; and
10 utilizing at least one of the first evaluation result and the second evaluation result as instructions, creating an output by comparing to at least one dataset cluster in the database thereby obtaining the corollary additive AI output result.
2. The method according to claim 1 further comprising measuring distance from the
15 additive AI result to an exemplar result from the dataset cluster to obtain an additive AI cluster identification.
3. The method according to claim 1, the dataset cluster further comprises matched
20 written templates.
4. The method according to claim 3 further comprising assembling the additive AI cluster identification and the matched written templates to obtain a report.
5. The method according to claim 4 further comprising, displaying the report to a user.
25
6. The method according to claim 1, the at least one second AI classifier is a derivative of the first AI classifier.
7. The method according to claim 6, the second AI classifier is trained using at least a
30 portion of data used for training the first AI classifier.
8. The method according to claim 6, the second AI classifier is related to first AI classifier.

9. The method according to claim 1, the first AI classifier is a comprehensive classifier.
10. The method according to claim 1, the second AI classifier is a specific classifier.
- 5 11. The method according to claim 1, the first AI classifier is a specific classifier.
12. The method according to claim 1, the second AI classifier is a comprehensive classifier.
- 10 13. The method according to claim 1 further comprising repeating the steps of directing and comparing for a series of daisy chained related or derivative AI classifiers.
14. The method according to claim 1 further comprising comparing the first evaluation
- 15 15. The method according to claim 1 further comprising adding the first evaluation result and the second evaluation result to a results database.
- 20 16. The method according to claim 1 further comprising creating database cluster entries using the at least one first AI classifier and one or more derivative classifiers of a first evaluation classifier.
- 25 17. The method according to claim 16, further comprising evaluating additional data inputs using the at least one first AI evaluation classifier and the one or more derivative classifiers of the first evaluation classifier to compare against the database cluster entries and return an exemplar result from the database.
- 30 18. The method according to claim 1 further comprising a system or user input that accepts at least one form of data inputs selected from: a pixel, collection of pixels, an image, a collection of images grouped together, a color, an arrangement of colors, a word, several words, a partial sentence, a complete sentence, a plurality of sentences together creating a

partial paragraph, a full paragraph, a plurality of paragraphs, a full template, an article, an essay or other text based output, a measurement, a number, a waveform, an equation, a recipe, and mixed data, for analysis and storage into the database or analysis and creation of an output report.

5

19. The method according to claim 1 further comprising prior to processing, acquiring the digital file.

10

20. The method according to claim 1 further comprising prior to processing, converting an analog file into the digital file.

15

21. The method according to claim 1 further comprising prior to processing, classifying the digital file by performing at least one of: labelling, cropping, editing, and orientating the digital file.

22. The method according to claim 1 comprising clustering of both AI derived and non-AI derived data.

20

23. The method according to claim 1 further comprising applying a standard mathematical formula, rearrangement or weighting to the AI result.

24. The method according to claim 1 further comprising prior to utilizing, obtaining and aggregating evaluation results in a database.

25

25. A system programmed to obtain an additive AI result by any of the methods according to claim 1, the system comprising:

at least one first AI processor;

at least one derivative AI processor that is derived from the first AI processor; and

an output device.

30

26. The system according to claim 25 further comprising at least one database library.

27. The system according to claim 25 further comprising a user interface.