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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 September 2020 (10.09.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/181098 A1

(51) International Patent Classification:
G06T 7/00 (2017.01)

(21) International Application Number:
PCT/US2020/021210

(22) International Filing Date:
05 March 2020 (05.03.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/814,230 05 March 2019 (05.03.2019) US

(71) Applicant: **MEMORIAL SLOAN KETTERING CANCER CENTER** [US/US]; 1275 York Avenue, New York, New York 10065 (US).

(72) Inventors: **FUCHS, Thomas J.**; c/o Memorial Sloan Kettering Cancer Center, 1275 York Avenue, New York, New York 10065 (US). **MUHAMMAD, Hassan**; c/o Memorial Sloan Kettering Cancer Center, 1275 York Avenue, New York, New York 10065 (US).

(74) Agent: **KHAN, Shabbi S.** et al.; Foley & Lardner LLP, 3000 K Street NW, Suite 600, Washington, District of Columbia 20007 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

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

(54) Title: SYSTEMS AND METHODS FOR IMAGE CLASSIFICATION USING VISUAL DICTIONARIES

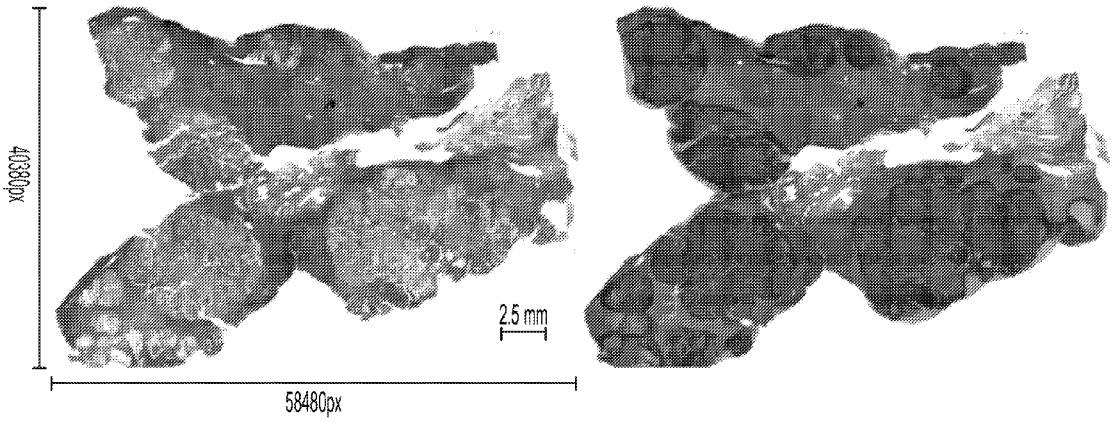

FIG. 1

(57) Abstract: Presented herein are systems and methods of clustering images using encoder-decoder models. A computing system may identify tiles derived from an image. Each tile may have a first dimension. The computing system may apply an image reconstruction model to the tiles. The image reconstruction model may include an encoder block having a first set of weights to generate embedding representations corresponding to the tiles. Each embedding representation may have a second dimension lower than the first dimension. The image reconstruction model may include a decoder block having a second set of weights to generate reconstructed tiles corresponding to the embedding representations. The computing system may apply a clustering model comprising a feature space to the embedding representations to classify each tile to one of a plurality of conditions.

WO 2020/181098 A1

[Continued on next page]

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

SYSTEMS AND METHODS FOR IMAGE CLASSIFICATION USING VISUAL DICTIONARIES

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS

[0001] The present application claims priority to U.S. Provisional Application No. 62/814,230, titled "SYSTEMS AND METHODS FOR IMAGE CLASSIFICATION USING VISUAL DICTIONARIES," filed March 5, 2019, which is incorporated herein by reference in its entirety.

BACKGROUND

[0002] A feature space representation may be derived for an image. A model or an algorithm executing on a computing device may be used to identify which classification the representation belongs to.

SUMMARY

[0003] At least one aspect of the present disclosure is directed to systems and methods of training encoder-decoder models to cluster images. A computing system having one or more processors coupled with memory may identify a plurality of tiles derived from a sample image of a training dataset. Each of the plurality of tiles may correspond to one of a plurality of conditions and have a first dimension. The computing system may apply an image reconstruction model to the plurality of tiles. The image reconstruction model may have an encoder block having a first set of weights to generate a plurality of embedding representations corresponding to the plurality of tiles. Each of the plurality of embedding representations may have a second dimension lower than the first dimension. The image reconstruction model may have a decoder block having a second set of weights to generate a plurality of reconstructed tiles corresponding to the plurality of embedding representations. Each of the plurality of reconstructed tiles may have a third dimension higher than the second dimension. The computing system may apply a clustering model comprising a feature space to the plurality of embedding representations to classify each the corresponding plurality of tiles to one of the plurality of conditions. The computing system may modify the feature

space of the clustering model based on classifying of the plurality of embedding representations to one of the plurality of conditions. The computing system may determine as first error metric between the plurality of tiles and the corresponding plurality of reconstructed tiles. The computing system may determine a second error metric based on classifying of the plurality of embedding representations to one of the plurality of conditions. The computing system may update at least one weight of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance with the first error metric and the second error metric.

[0004] In some embodiments, the computing system may identify a plurality of centroids defined by the feature space of the clustering model prior to applying of the clustering model to the plurality of embedding representations, the plurality of centroids corresponding to the plurality of conditions. The plurality of centroids may correspond to the plurality of conditions. In some embodiments, the computing system may identify a plurality of points defined within the feature space of the clustering model for the corresponding plurality of embedding representations. In some embodiments, the computing system may determine the second error metric between the plurality of centroids and the plurality of points.

[0005] In some embodiments, the computing system may determine a combined error metric in accordance with a weighted summation of the first error metric and the second error metric. In some embodiments, the computing system may update the at least one of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance to the combined error metric.

[0006] In some embodiments, the computing system may determine that the clustering model is not at a convergence state based on a comparison of a movement metric for a plurality of centroids in the feature space to a threshold value. In some embodiments, the computing system may reapply the image reconstruction model to the plurality of tiles responsive to determining that clustering model is not at the convergence state.

[0007] In some embodiments, the computing system may apply, subsequent to updating the at least one weight, the image reconstruction to the plurality of tiles. The

encoder block may generate a second plurality of embedding representations corresponding to the plurality of tiles. In some embodiments, the computing system may apply the clustering model to the second plurality of embedding representations to classify at least one of the plurality of tiles to a first condition of the plurality of conditions different from a second condition of the plurality of conditions as classified prior to the modifying of the feature space.

[0008] In some embodiments, the computing system may initialize the clustering model comprising the feature space to define a plurality of centroids corresponding to the plurality of conditions . Each of the plurality of centroids may be at least one of a random point or a predefined point within feature space.

[0009] In some embodiments, the computing system may identify, from the sample image, a region of interest corresponding to one of the plurality of conditions labeled by an annotation of the training dataset for the sample image. In some embodiments, the computing system may generate, using the region of interest identified from the sample image, the plurality of tiles from the sample image.

[0010] In some embodiments, the computing system may identify the plurality of tiles derived from the sample image of the training dataset. The sample image may be derived from a tissue sample via a histopathological image preparer. The sample image may include a region of interest corresponding to one of the plurality of conditions present in the tissue sample.

[0011] At least one aspect of the present disclosure is directed to systems and methods of clustering images using encoder-decoder models. A computing system having one or more processors coupled with memory may identify a plurality of tiles derived from an image acquired via an image acquisition device. Each of the plurality of tiles may have a first dimension. The computing system may apply an image reconstruction model to the plurality of tiles. The image reconstruction model may include an encoder block having a first set of weights to generate a plurality of embedding representations corresponding to the plurality of tiles. Each of the plurality of embedding representations may have a second dimension lower than the first dimension. The image reconstruction model may include a

decoder block having a second set of weights to generate a plurality of reconstructed tiles corresponding to the plurality of embedding representations. The computing system may apply a clustering model comprising a feature space to the plurality of embedding representations to classify each the corresponding plurality of tiles to one of a plurality of conditions.

[0012] In some embodiments, the computing system may apply clustering model comprising the feature space. The feature space may define a plurality of centroids corresponding to the plurality of conditions to classify each of the corresponding plurality of tiles to one of the plurality of conditions. In some embodiments, the computing system may identify, from the image, a region of interest corresponding to one of the plurality of conditions. In some embodiments, the computing system may generate, using the region of interest identified from the image, the plurality of tiles from the image.

[0013] In some embodiments, the computing system may identify the plurality of tiles derived from the image. The image may be derived from a tissue sample via a histopathological image preparer. The image may include a region of interest corresponding to one of the plurality of conditions present in the tissue sample. In some embodiments, the computing system may train the image reconstruction model to modify at least one weight of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance with an error metric determined using the clustering algorithm.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The foregoing and other objects, aspects, features, and advantages of the disclosure will become more apparent and better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

[0015] FIG. 1 depicts screenshots of whole slide images: On the left is an example of a digitized slide at low magnification. These slides are quite large, as this average example comprises at least one billion pixels of tissue. On the right, the same slide with regions of tumor annotated. Note that annotation may cover areas of background. The novel tiling

protocol with quality control ensures that tiles contain high resolution and clear images of tissue.

[0016] FIG. 2 depicts screenshots of tiles from biomedical images. Randomly selected examples of tiles excluded by the quality control algorithm. Interestingly, this method also helped identify tiles with ink stains, folds, and those which include partial background, tiles which the first stage of tile generation was designed to exclude as much as possible.

[0017] FIG. 3 depicts a block diagram of an auto-encoder model. At each iteration, the model is updated in two steps. After each forward-pass of a minibatch, the network weights are updated. At the end of each epoch, centroid locations are updated by reassigning all samples in the newly updated embedding space to the nearest centroid from the previous epoch, as described in equation 3. Finally, each centroid location is recalculated using equation 4. All centroids are randomly initialized before training.

[0018] FIG. 4 depicts graphs of Multivarite Cox regression comprising all clusters as covariates. Clusters 0, 11, 13, 20, and 23 show significant hazard ratios. Log-Rank Test was used to measure significance of the entire model.

[0019] FIG. 5 depicts graphs of Kaplan-Meier survival curves. The top panels show Kaplan-Meier survival curves across time (months) for clusters 0, 11, and 13 with reported stratification significance based on the Log-Rank Test. The middle panel shows the amount of samples in each stratified class over time and the bottom panel indicates points at which censored events occur. Each analysis shows a significantly positive prognostic factor for samples positive for the given cluster.

[0020] FIG. 6 depicts a screenshot of sampled tiles. Each row depicts 20 randomly sampled tiles for each cluster.

[0021] FIG. 7 depicts a block diagram of an auto-encoder model. Training the model is the fourth phase of the complete pipeline. At each iteration, the model is updated in two steps. After each forward-pass of a minibatch, the network weights are updated. At the end of each epoch, centroid locations are updated by reassigning all samples in the newly updated

embedding space to the nearest centroid from the previous epoch, as described in Eq. 3. Finally, each centroid location is recalculated using Eq. 4. All centroids are randomly initialized before training.

[0022] FIG. 8 depicts a graph of Kaplan-Meier visualization of survival probabilities for each patient classified into one of the five cluster classes produced by the unsupervised model. Patients with high presence of tissue in cluster 3 have a better recurrence-free survival than those with clusters 2 or 4.

[0023] FIG. 9A depicts a block diagram of a system for clustering images using autoencoder models.

[0024] FIG. 9B depicts a block diagram of models in the system for clustering images using autoencoder models.

[0025] FIG. 9C depicts a block diagram of an encoder block of an image reconstruction model in the system for clustering images.

[0026] FIG. 9D depicts a block diagram of a convolution stack in an encoder block of an image reconstruction model in the system for clustering images.

[0027] FIG. 9E depicts a block diagram of a decoder block of an image reconstruction model in the system for clustering images.

[0028] FIG. 9F depicts a block diagram of a deconvolution stack in an encoder block of an image reconstruction model in the system for clustering images

[0029] FIG. 9G depicts a block diagram of a clustering model in the system for clustering images.

[0030] FIG. 10A depicts a flow diagram of a method of training autoencoder models to cluster images.

[0031] FIG. 10B depicts a flow diagram of a method of clustering images using autoencoder models.

[0032] FIG. 11 depicts a block diagram of a server system and a client computer system in accordance with an illustrative embodiment.

DETAILED DESCRIPTION

[0033] Following below are more detailed descriptions of various concepts related to, and embodiments of, systems and methods for maintaining databases of biomedical images. It should be appreciated that various concepts introduced above and discussed in greater detail below may be implemented in any of numerous ways, as the disclosed concepts are not limited to any particular manner of implementation. Examples of specific implementations and applications are provided primarily for illustrative purposes.

[0034] Section A describes unsupervised cancer subtyping using a histological visual dictionary.

[0035] Section B describes unsupervised subtyping of cholangiocarcinoma using a deep clustering convolutional autoencoder.

[0036] Section C describes systems and methods for clustering images using autoencoder models.

[0037] Section D describes a network environment and computing environment which may be useful for practicing various embodiments described herein.

A. Unsupervised Cancer Subtyping Using a Histological Visual Dictionary

[0038] Unlike common cancers, such as those of the prostate and breast, tumor grading in rare cancers is difficult and largely undefined because of small sample sizes and the sheer volume of time to embark on such a task. One of the most challenging examples is intrahepatic cholangiocarcinoma (ICC), for which there is well-recognized tumor heterogeneity and no grading paradigm or prognostic biomarkers.

[0039] Presented herein is a new deep convolutional autoencoder-based clustering model that, without supervision, groups together cellular and structural morphologies of tumor in 246 ICC digitized whole slides, based on visual similarity. From this visual

dictionary of histologic patterns, the clusters can be used as covariates to train Cox-proportional hazard survival models. In a univariate analysis, these three clusters showed high significance for recurrence-free survival, as well as multivariate combinations of them. In a multivariate analysis of all clusters, five showed high significance to recurrence-free survival, however the overall model was not measured to be significant. Finally, a pathologist assigned clinical terminology to the prognosis-correlated clusters in the visual dictionary and found evidence supporting the hypothesis that collagen-enriched fibrosis plays a role in disease severity. These results offer insight into the future of cancer subtyping and show that computational pathology can contribute to disease prognostication, including rare cancers.

1. Introduction

[0040] Following below are more detailed descriptions of various concepts related to, and embodiments of, systems and methods for maintaining databases of biomedical images. It should be appreciated that various concepts introduced above and discussed in greater detail below may be implemented in any of numerous ways, as the disclosed concepts are not limited to any particular manner of implementation. Examples of specific implementations and applications are provided primarily for illustrative purposes. Cancer grading is an important tool used to predict disease prognosis and direct therapy. Commonly occurring cancers, such as those of breast and prostate, have well established grading schemes validated on large sample sizes. The grading system for prostatic adenocarcinoma is the foremost internationally accepted grading scheme in cancer medicine. The Gleason Grading System (GGS) relates observed histologic patterns in pathology to outcome data using thousands of cases. After nearly twenty years of repeatedly designing and validating his prognostic classifications, a final review of grading prostate cancer was published. Since then, it has been a clinical standard. Although the gold standard in prostate cancer stratification, GGS is subject to ongoing critical assessment. A new dictionary of histologic patterns was created, independent from those which constituted the GGS, from a cohort of 1275 cases followed over five years. Using these specific patterns, significantly stratified risk groups was found within GGS grades, supporting the notion that GGS can be further optimized. The manual labor required to identify different histologic patterns and use them to stratify patients into

different risk groups is an extremely complex task requiring years of effort and repeat reviews of large amounts of visual data, often by one pathologist. The story of prostate carcinoma grading indicates different observers may identify different or incomplete sets of patterns.

[0041] Developing a grading system for a rare cancer poses a unique set of challenges. Intrahepatic cholangiocarcinoma (ICC), a cancer of the bile duct, has an incidence of 1 in 100,000 in the United States. There exists no universally accepted histopathology-based subtyping or grading system for ICC and studies classifying ICC into different risk groups have been inconsistent. A major limiting factor to subtyping ICC is that only small cohorts are available to each research institution. A study using one of the world's largest cohorts of ICC (n= 184) was performed, expanding one proposed histology-based binary subtyping into four risk groups but still found no significant stratification.

[0042] There is an urgent need for efficient identification of prognostically relevant cellular and structural morphologies from limited histology datasets of rare cancers to build risk stratification systems which are currently lacking across many cancer types. Ideally, these systems should utilize a visual dictionary of histologic patterns that is comprehensive and reproducible. Once generated, such a visual dictionary is to be translatable to histopathological terms universally understood by pathologists. Computational pathology offers a new set of tools, and more importantly, a new way of approaching the historical challenges of subtyping cancers using computer vision-based deep learning, leveraging the digitization of pathology slides, and taking advantage of the latest advances in computational processing power. Presented herein is a new deep learning-based model which can create such a visual dictionary and show utility by stratifying ICC, based on morphology at the cellular level for the first time.

2 Materials and Methods

[0043] Cancer histopathology images exhibit high intra- and inter-heterogeneity because of their size (as large as tens of billions of pixels). Different spatial or temporal sampling of a tumor can have sub-populations of cells with unique genomes, theoretically resulting in visually different patterns of histology. In order to effectively cluster this

extremely large amount of high intra-variance data into subsets which are based on similar morphologies, a neural network-based clustering cost-function may be combined with a novel deep convolutional architecture. The cost-function may outperform other clustering techniques on images of hand-written digits. Finally, the power and usefulness of this clustering model may be assessed by conducting survival analysis, using both Cox-proportional hazard modeling and Kaplan-Meier survival estimation, to measure if each cluster of histomorphologies has significant correlation to recurrence of cancer after resection.

2.1 Deep Clustering Convolutional Auto Encoder

[0044] A convolutional auto-encoder is made of two parts, an encoder and decoder. The encoder layers project an image into a lower dimensional representation, an embedding, through a series of convolution, pooling, and activation functions. This is described in equation 1a, where x_i is an input image or input batch of images transformed by $f_\theta()$, and z_i is the resulting representation embedding. The decoder layers try to reconstruct the original input image from its embedding using similar functions. Mean-squared-error loss (MSE) is commonly used to optimize such a model, updating model weights (θ) relative to the error between the original (input, x_i) image and the reconstruction (output, x_i') image in a set of N images. This is shown in equation 1b.

$$(a)\ z_i = f_\theta(x_i) \qquad (b)\ \in = \min_\theta \frac{1}{N}\sum_{i=1}^{N}\|x_i - x_i'\|^2 \tag{1}$$

[0045] Although a convolutional auto-encoder can learn effective lower-dimensional representations of a set of images, it does not cluster together samples with similar morphology. To overcome this problem, the MSE-loss function may be modified by using the reconstruction-clustering error function:

$$\in = \min_\theta \frac{1}{N}\sum_{i=1}^{N}\|x_i - x_i'\|^2 + \lambda \sum_{i=1}^{N}\|z_i - c_i^*\|^2, \tag{2}$$

where z_i is the embedding as defined in equation 1a, c_i^* is the centroid assigned to sample x_i from the previous training epoch, and λ is a weighting parameter. Cluster assignment is

determined by finding the shortest Euclidean distance between a sample embedding from epoch t and a centroid, across j centroids from epoch $t-1$:

$$c_i^* = \arg\min_j \left\| z_i^t - c_j^{t-1} \right\|^2 \quad (3)$$

[0046] The algorithm is initialized by assigning a random cluster to each sample. Centroid locations are calculated for each cluster class by equation 4. Each mini-batch is forwarded through the model and network weights are respectively updated. At the end of an epoch, defined by the forward-passing of all mini-batches, cluster assignments are updated by equation 3, given the new embedding space. Finally, the centroid locations are updated from the new cluster assignments. This process is repeated until convergence. FIG. 3 shows a visualization of this training procedure.

$$c_j^t = \frac{\sum_{t=1}^{N} z_i}{\left| c_j^{t-1} \right|} \quad (4)$$

2.2 Dataset

[0047] Two hundred forty six whole slides from patients with resected ICC without neoadjuvant chemotherapy were acquired with approval. These slides were digitized using Aperio AT2 scanners. Up-to-date retrospective data for recurrence free survival after resection was also obtained. Though currently a small sample size, this collection is the largest known retrospective ICC dataset in the world.

[0048] A library of extracted image tiles was generated from all digitized slides. First, each slide was reduced to a thumbnail, where one pixel in the thumbnail represented a 224 x 224 px tile in the slide at 20x magnification. Next, using Otsu thresholding on the thumbnail, a binary mask of tissue (positive) vs. background (negative) was generated. Connected components below 10 thumbnail pixels in tissue were considered background to exclude dirt or other insignificant masses in the digitized slide. Finally, mathematical morphology was used to erode the tissue mask by one thumbnail pixel to minimize tiles with partial background. To separate the problem of cancer subtyping, as discussed in this paper, from the problem of tumor segmentation, the areas of tumor were manually annotated using a

web-based whole slide viewer. Using a touchscreen, a pathologist painted over regions of tumor to identify where tiles should be extracted. FIG. 1 illustrates an example of this annotation. Tiles were added to the training set if they lay completely within these regions of identified tumor.

2.2.1 Quality Control

[0049] Scanning artifacts such as out-of-focus areas of an image can impact model performance on smaller datasets. A deep convolutional neural network was trained to detect blurred tiles to further reduce noise in the dataset. Training a detector on real blur data was beyond the scope of this study because obtaining annotations for blurred regions in the slide is unfeasible and would also create a strong class imbalance between blurred and sharp tiles. To start, half of the tiles were artificially blurred by applying a Gaussian-blur filter with a random filter radius ranging from 1 to 10. The other half were labeled "sharp" and no change was made to them. A ResNet18 was trained to output an image quality score by regressing over the values of the applied filter radius using MSE. A value of 0 was used for images in the sharp class. Finally, a threshold was manually selected to exclude blurred images based on the output value from the detector. FIG. 2 shows randomly selected examples of tiles excluded based on the blur detector.

2.3 Architecture and Training

[0050] Presented herein is a novel convolutional autoencoder architecture to optimize performance in image reconstruction. The encoder is a ResNet18 which was pretrained on ImageNet. The parameters of all layers of the encoder updated when training the full model on pathology data. The decoder is comprised of five convolutional layers, each with a padding and stride of 1, for keeping the tensor size constant with each convolution operation. Upsampling is used before each convolution step to increase the size of the feature map. Empirically, batch normalization layers did not improve reconstruction performance and thus, were excluded.

[0051] Two properties of the model are to be optimized: first, the weights of the network, θ, and then locations of the cluster centers, or centroids, in the embedding space, C_j.

In order to minimize equation 2 and update θ, the previous training epoch's set of centroids, C_j^{t-1}, is used. In the case of the first training epoch, centroid locations are randomly assigned upon initialization. A training epoch is defined by the forward-passing of all mini-batches once through the network. After θ have been updated, all samples are reassigned to the nearest centroid using equation 3. Finally, all centroid locations are updated using equation 4 and used in the calculations of the next training epoch. FIG. 3 illustrates this process and architecture.

[0052] All training was done on DGX-1 compute nodes (NVIDIA Corp., Santa Clara, CA) using PyTorch 0.4 on Linux CentOS7. The model was trained using Adam optimization for 125 epochs, a learning rate of $1e^{-2}$, and weight decay of $1e^{-4}$. The learning rate was decreased by 0.1 every 50 epochs. The clustering weight, λ, was set to 0.4. Finally, to save on computation time, 500,000 tiles were randomly sampled from the complete tile library to train each model, resulting in approximately 2000 tiles from each slide on average.

2.3.1 Survival Analysis

[0053] In order to measure the usefulness and effectiveness of the clustered morphological patterns, slide-level survival analysis may be conducted, based on which patterns occurred on a given digital slide to its associated outcome data. Each cluster was considered a binary covariate. If one or more tile(s) from a given cluster existed in a slide, the slide was considered positive for the morphological pattern defined by that cluster. A multivariate Cox regression was used to model the impact of all clusters on recurrence based on this binarized representation of each cluster for each patient:

$$H(t) = h_o e^{b_1 x_1 + b_2 x_2 + \cdots + b_j x_j}, \quad (5)$$

where $H(t)$ is the hazard function dependent on time t, h_o is a baseline hazard, and covariates (x_1, x_2, ... x_j) have coefficients (b_1, b_2, ... b_j). A covariate's hazard ratio is defined by e^{b_j}. A hazard ratio greater than one indicates that tiles in that cluster contribute to a worse prognosis. Conversely, a hazard ratio less than one contributes to a good prognostic factor. To measure significance in the survival model, p-values based on the Wald statistic are presented for each covariate.

[0054] A univariate Cox Regression was also performed on each cluster. Those measured as significant ($p < 0.05$) were used to build multivariate Cox regressions for each combination. The results are described in Table 2. Finally, Kaplan-Meier curves of the prognositically significant clusters may be shown by estimating the survival function S(*t*):

$$S(t) = \prod_{t_i<t} \frac{n_i - d_i}{n_n}, \quad (6)$$

where d_i are the number of recurrence events at time t and n_i are the number of subjects at risk of death or recurrence prior to time t. This binary Kaplan-Meier analysis was done for each cluster, and stratification was measured to be significant using a standard Log-Rank Test.

2.3.2 Histologic Interpretation

[0055] Clusters measured to be significantly correlated with survival based on the Cox analysis were assigned clinical descriptors by a pathologist using standard histological terminology as shown in table 1. For 20 random tiles of each of those clusters, a semi-quantitative assessment of histological elements comprising each tile was recorded. A major feature was defined as presence of a histological element in >50 percent of a single tile area in greater than 10 tiles of a cluster. Minor features were defined as histological elements in > 50 percent of the tile area in 8-9 tiles of a cluster. A major tumor histology type was defined for a cluster when >50 percent of the tiles containing any amount of tumor were of the same histologic description.

{0056} Table 1: Standard pathology terms were used to build a description of the most common histologic elements appearing in each cluster.

Category	Histologic Description	
Debris	Granular fibrinoid material, amorphous	
	Granular fibrinoid material, ghost cells	
	Granular fibrinoid material, pkynotic nuclei	
	Nectroic tumor	
	Red blood cells	
Extracellular Matrix	Collagen, linear fascicles	
	Collagen, wavy fascicles	
	Collagen, bundles in cross section	
	Collagen, amorphous	
	Mucin	
Hematolymphoid	Neutrophils	
	Lymphocytes	
	Histiocytes	
Other Non-Neoplastic Elements	Vessel	
	Nerve	
	Hepatocytes	
	Fibroblasts	
Tumor Histology Type	Tubular	High nuclear:cytolasmic ratio
		Low nuclear:cytolasmic ratio
	Solid	High nuclear:cytolasmic ratio

	Low nuclear:cytolasmic ratio
Too limited to classify	

2.4 Results

[0057] The multivariate Cox model comprising all clusters showed significance in the hazard ratios of clusters 0, 11, 13, 20, and 23 ($p < 0.05$). Cluster 20 showed a decrease in prognostic risk and clusters 0, 11, 13, and 23 showed an increase in prognostic risk. However, the overall model was not measured to be significant (Likelihood Ratio Test: $p = 0.106$, Wald Test: $p = 0.096$, Log-Rank Test: $p = 0.076$).

[0058] Clusters 0, 11, and 13 were measured to be significant ($p < 0.05$) by the univariate Cox Regression, all with a positive influence in prognosis when compared to samples negative for those clusters. Table 2 shows the individual univariate hazard ratios for Clusters 0, 11, 13 and all combinations for a multivariate Cox regression when considering only those significant clusters. For these multivariate models, the Wald Test, Likelihood Ratio Test, and Log-Rank Test, all showed significance of $p < 0.05$.

[0059] For the significant clusters from the univariate analysis, FIG. 5 shows Kaplan-Meier plots for patients stratified into positive and negative groups. A Log-Rank Test p-value of less than 0.05 shows significance in stratification in the estimated survival curves. Each vertical tick indicates a censored event.

[0060] Semi-quantitative histologic analysis of the random 20 tiles selected from the clusters of significance showed that only cluster 0 met criteria for a major feature, consisting of the extracellular matrix component, collagen, specifically arranged in linear fascicles. Collagen was a minor feature for one other cluster (23) and both of these clusters (0, 23) had an decrease in hazard ratio on univariate survival analysis. No tumor histology, as defined in table 1, met the criterion as a major feature. One tumor histology was a minor feature of one cluster, clusters 13 had 9 tiles with more than 50 percent solid tumor histology with low nuclear:cytoplasmic ratio and this cluster indicated a decrease in prognostic risk. No other major or minor features were identified.

[0061] Although tumor content was not a major or minor feature of most clusters, tumor content of any volume and histologic description was present in 35-75 percent of tiles for each cluster. Major tumor histology types were seen in two clusters: cluster 0 had 4/7 (57 percent) of tiles containing tubular type, and cluster 23 had 7/12 (58 percent) of tiles containing tubular high nuclear:cytoplasmic ratio type.

[0062] Table 2: Hazard ratios (e^{b_j}) for prognositically significant clusters j when modelling a univariate Cox regression and their combinations in multivariate models. The values in parenthesis indicate bounds for 95% confidence intervals based on cumulative hazard.

	Univariate	Multivariate			
		0+11	0+13	11+13	0+11+13
Cluster 0	0.618*** (0.447 – 0.855)	0.644*** (0.463 – 0.895)	0.675** (0.459 – 0.993)		0.725 (0.489 – 1.075)
Cluster 11	0.515** (0.280 – 0.951)	0.598 (0.320 – 1.116)		0.494** (0.267 – 0.915)	0.560* (0.297 – 1.056)
Cluster 13	0.750* (0.517 – 0.961)		0.855 (0.591 – 1.127)	0.694** (0.508 – 0.946)	0.813 (8.561 – 1.178)
Wald Test		11.36***	9.15**	9.75***	12.51***
Likelihood Ratio Test		10.19***	8.59**	8.86**	11.37***
Score (Log-Rank) Test		11.67***	9.31***	9.99***	12.85***
Note:	Significance codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ‘ 1				

2.5 Conclusion

2.5.1 Technical Contribution

[0063] This model offers a novel approach for identifying histological patterns of potential prognostic significance, circumventing the tasks of tedious tissue labeling and laborious human evaluation of multiple whole slides. As a point of comparison, a recent study showed that an effective prognostic score for colorectal cancer was achieved by first segmenting a slide into eight predefined categorical regions using supervised learning. Methods such as this limit the model to pre-defined histologic components (tumor, non-tumor, fat, debris, etc) and the protocol may not extend to extra-colonic anatomic sites lacking a similar tumor specific stroma interactions. In contrast, the design of the model lacks predefined tissue classes, and has the capability to analyze n clusters, thus removing potential bias introduced by training and increasing flexibility in application of the model.

2.5.2 Histology

[0064] By semi-quantitative assessment of the histologic components of the tiles in clusters with prognostic significance, tumor cells were discovered to be not a major feature in any cluster, whereas two clusters had connective tissue (stroma) comprised of extracellular matrix (collagen) as a major/minor feature.

[0065] Tumor stroma, the connective tissue intervening between tumor cell clusters in tissue comprised of matrix and collagen, is known to play an integral role in cancer growth, angiogenesis, and metastasis, but is not used in tumor grading or prognostication systems, which tend to focus on tumor attributes such as nuclear features, cellular architecture, and invasive behavior. Research specifically in ICC has supported the important biological role of tumor associated stroma in tumor progression by analyzing unseen paracrine factors. Recently, a deep learning- based algorithm used tumor associated stroma, not tumor cells, to stratify ductal carcinoma in situ of the breast by grade. In the present study, stroma was found to be a major/minor feature of two significant clusters which this raises the possibility that the stroma microenvironment could have distinct morphologic characteristics that could be detectable routinely and potentially prognostically significant.

B. Unsupervised Subtyping of Cholangiocarcinoma Using a Deep Clustering Convolutional Autoencoder

[0066] Unlike common cancers, such as those of the prostate and breast, tumor grading in rare cancers is difficult and largely undefined because of small sample sizes, the sheer volume of time and experience needed to undertake such a task, and the inherent difficulty of extracting human-observed patterns. One of the most challenging examples is intrahepatic cholangiocarcinoma (ICC), a primary liver cancer arising from the biliary system, for which there is well-recognized tumor heterogeneity and no grading paradigm or prognostic biomarkers. Presented herein is a new unsupervised deep convolutional autoencoder-based clustering model that groups together cellular and structural morphologies of tumor in 246 digitized whole slides, based on visual similarity. Clusters based on this visual dictionary of histologic patterns are interpreted as new ICC subtypes and evaluated by training Cox-proportional hazard survival models, resulting in statistically significant patient stratification.

1. Introduction

[0067] Cancer subtyping is an important tool used to determine disease prognosis and direct therapy. Commonly occurring cancers, such as those of breast and prostate, have well established subtypes, validated on large sample sizes. The manual labor required to subtype a cancer, by identifying different histologic patterns and using them to stratify patients into different risk groups, is an extremely complex task requiring years of effort and repeat review of large amounts of visual data, often by one pathologist.

[0068] Subtyping a rare cancer poses a unique set of challenges. Intrahepatic cholangiocarcinoma (ICC), a primary liver cancer emanating from the bile duct, has an incidence of approximately 1 in 160,000 in the United States, and rising. Currently, there exists no universally accepted histopathology-based subtyping or grading system for ICC and studies classifying ICC into different risk groups have been inconsistent. A major limiting factor to subtyping ICC is that only small cohorts are available to each institution. There is an urgent need for efficient identification of prognostically relevant cellular and structural morphologies from limited histology datasets of rare cancers, such as ICC, to build risk

stratification systems which are currently lacking across many cancer types. Computational pathology offers a new set of tools, and more importantly, a new way of approaching the historical challenges of subtyping cancers using computer vision-based deep learning, leveraging the digitization of pathology slides, and taking advantage of the latest advances in computational processing power. Presented herein is a new deep learning-based model which uses a unique neural network-based clustering approach to group together histology based on visual similarity. With this visual dictionary, clusters may be interpreted as subtypes and train a survival model, showing significant results for the first time in ICC.

2. Materials and Methods

[0069] Cancer histopathology images exhibit high intra- and inter-heterogeneity because of their size (as large as tens of billions of pixels). Different spatial or temporal sampling of a tumor can have sub-populations of cells with unique genomes, theoretically resulting in visually different patterns of histology. In order to effectively cluster this extremely large amount of high intra-variance data into subsets which are based on similar morphologies, a neural network-based clustering cost-function, shown to outperform clustering techniques on images of hand-written digits, may be combined with a novel deep convolutional architecture. It is hypothesized that a k-means style clustering cost function under the constraint of image reconstruction which is being driven by adaptive learning of filters will produce clusters of histopathology relevant to patient outcome. Finally, the performance and usefulness of this clustering model may be assessed by conducting survival analysis, using both Cox-proportional hazard modeling and Kaplan-Meier survival estimation, to measure if each cluster of histomorphologies has significant correlation to recurrence of cancer after resection. While other studies have performed unsupervised clustering of whole slide tiles based on image features, they have been used to address the problem of image segmentation and relied on clustering a developed latent space. This study adjusts the latent space with each iteration of clustering.

2.1 Deep Clustering Convolutional Autoencoder

[0070] A convolutional auto-encoder is made of two parts, an encoder and decoder. The encoder layers project an image into a lower dimensional representation, an embedding,

through a series of convolution, pooling, and activation functions. This is described in Eq. 1a, where x_i is an input image or input batch of images transformed by $f_\theta()$, and z_i is the resulting representation embed- ding. The decoder layers try to reconstruct the original input image from its embedding using similar functions. Mean-squared-error loss (MSE) is commonly used to optimize such a model, updating model weights (θ) relative to the error between the original (input, x_i) image and the reconstruction (output, x_i') image in a set of N images. This is shown in Eq. 1b.

$$(a)\ z_i = f_\theta(x_i) \qquad (b)\ \epsilon = \min_\theta \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \|x_i - x_i'\|^2 \qquad (1)$$

[0071] Although a convolutional auto-encoder can learn effective lower-dimensional rep- resentations of a set of images, it does not cluster together samples with similar morphology. To overcome this problem, the reconstruction-clustering error function may be used:

$$\epsilon = \min_\theta \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \|x_i - x_i'\|^2 + \lambda \sum_{i=1}^{\mathrm{N}} \|z_i - c_i^*\|^2, \qquad (2)$$

[0072] where z_i is the embedding as defined in Eq. 1a, c_i^* is the centroid assigned to sample x_i from the previous training epoch, and λ is a weighting parameter. Cluster assignment is determined by finding the shortest Euclidean distance between a sample embedding from epoch t and a centroid, across j centroids from epoch $t-1$:

$$c_i^* = \arg\min_j \left\| z_i^t - c_j^{t-1} \right\|^2 \qquad (3)$$

[0073] The algorithm is initialized by assigning a random cluster to each sample. Centroid locations are calculated for each cluster class by Eq. 4. Each mini-batch is forwarded through the model and network weights are respectively updated. At the end of an epoch, defined by the forward-passing of all mini-batches, cluster assignments are updated by Eq. 3, given the new embedding space. Finally, the centroid locations are updated from the new cluster assignments. This process is repeated until convergence. FIG. 7 shows a visualization of this training procedure.

$$c_j^t = \frac{\sum_{t=1}^{N} z_i}{\left|c_j^{t-1}\right|} \quad (4)$$

2.2 Dataset

[0074] Whole slide images were obtained. In total, 246 patients with resected ICC without neoadjuvant chemotherapy were included in the analysis. All slides were digitized using Aperio AT2 scanners (Leica Biosystems; Wetzlar Germany). Up- to-date retrospective data for recurrence free survival after resection were also obtained. Though currently a small sample size when compared to commonly occurring cancers, this collection is the largest known ICC dataset in the world.

[0075] A library of extracted image tiles was generated from all digitized slides. First, each slide was reduced to a thumbnail, where one pixel in the thumbnail represented a 224 224px tile in the slide at 20x magnification. Next, using Otsu thresholding on the thumbnail, a binary mask of tissue (positive) vs. background (negative) was generated. Connected components below 10 thumbnail pixels in tissue were considered background to exclude dirt or other insignificant masses in the digitized slide. Finally, mathematical morphology was used to erode the tissue mask by one thumbnail pixel to minimize tiles with partial background. To separate the problem of cancer subtyping, as discussed in this paper, from the problem of tumor segmentation, the areas of tumor were manually annotated using a web-based whole slide viewer. Using a touchscreen, a pathologist painted over regions of tumor to identify where tiles should be extracted. Tiles were added to the training set if they lay completely within these regions of identified tumor.

[0076] **Quality Control**. Scanning artifacts such as out-of-focus areas of an image can impact model performance on smaller datasets. A deep convolutional neural net- work was trained to detect blurred tiles to further reduce noise in the dataset. Training a detector on real blur data was beyond the scope of this study because obtaining annotations for blurred regions in the slide is unfeasible and would also create a strong class imbalance between blurred and sharp tiles. The data may be prepared for training a blur detector. To start, half of the tiles were artificially blurred by applying Gaussian-blur with a random radius ranging from 1 to 10. The other half were labeled “sharp” and no change was made to them. A

ResNet18 was trained to output an image quality score by regressing over the values of the applied filter radius using MSE. A value of 0 was used for images in the sharp class. Finally, a threshold was manually selected to exclude blurred tiles based on the output value of the detector.

2.3 Architecture and Training

[0077] Presented herein is a novel convolutional autoencoder architecture to optimize performance in image reconstruction. The encoder is a ResNet18 pretrained on ImageNet. The parameters of all layers of the encoder updated when training the full model on pathology data. The decoder is comprised of five convolutional layers, each with a padding and stride of 1, for keeping the tensor size constant with each convolution operation. Upsampling is used before each convolution step to increase the size of the feature map. Empirically, batch normalization layers did not improve reconstruction performance and thus, were excluded.

[0078] Two properties of the model need to be optimized: first, the weights of the network, θ, and then locations of the cluster centers, or centroids, in the embedding space, C_j. In order to minimize Eq. 2 and update θ, the previous training epoch's set of centroids, C_j^{t-1}, is used. In the case of the first training epoch, centroid locations are randomly assigned upon initialization. A training epoch is defined by the forward-passing of all mini-batches once through the network. After θ have been updated, all samples are reassigned to the nearest centroid using Eq. 3. Finally, all centroid locations are updated using Eq. 4 and used in the calculations of the next training epoch. FIG. 7 illustrates this process and architecture.

[0079] All training was done on DGX-1 compute nodes (NVIDIA, Santa Clara, CA) using PyTorch 0.4 on Linux CentOS 7. The models were trained using Adam optimization for 150 epochs, a learning rate of $1e^{-2}$, and weight decay of $1e^{-4}$. To save on computation time, 100,000 tiles were randomly sampled from the complete tile library to train each model, resulting in approximately 400 tiles from each slide on average. The following section describes the selection process for hyper-parameters λ and J, clustering weight and number of clusters, respectively.

[0080] Experiments. The Calinski-Harabaz Index (*CHI*), also known as the variance ratio criterion, was used to measure cluster performance, defined by measuring the ratio of between-clusters dispersion mean and within-cluster dispersion. A higher *CHI* indicates stronger cluster separation and lower variance within each cluster.

[0081] A series of experiments were conducted to optimize λ and J for model selection. With λ set to 0.2, five different models were trained with varying J clusters, ranging from 5 to 25. Secondly, five models were trained with varying λ, from 0.2 to 1, with J set to the value corresponding with the highest CHI from the previous experiment. A final model was trained with optimized J and λ to cluster all tiles in the dataset. Each slide was assigned a class based on which cluster was measured to occupy the largest area in the slide. This approach is used because it is similar to how a pathologist would classify a cancer into a subtype based on the most commonly occurring histologic pattern.

[0082] **Survival Modeling**. In order to measure the usefulness and effectiveness of the clustered morphological patterns, slide-level survival analysis may be conducted, based on the assigned classes to associated outcome data. Survival data often includes right-censored time durations. This means that the time of event of interest, in this case recurrence detected by radiology, is unknown for some patients. However, the time duration of no recurrence, as measured until the patient's last follow-up date, is still useful information which can be harnessed for modeling. Cox-proportional hazard modeling is a commonly used model to deal with right-censored data:

$$H(t) = h_o e^{b_i x_i}, \qquad (5)$$

where $H(t)$ is the hazard function dependant on time t, h_o is a baseline hazard, and covariate x_i is weighted by coefficient b_i. The hazard ratio, or relative likelihood of death, is defined by e^{b_j}. A hazard ratio greater than one indicates that a cluster class contributes to a worse prognosis. Conversely, a hazard ratio less than one contributes to a good prognostic factor. To measure significance in the survival model, p-values based on the Wald statistic, likelihood ratio, and log-rank test are presented for each model.

[0083] Five univariate cox models were trained, each with one cluster class held out as a reference to measure impact of survival relative to the other classes. Further, Kaplan-Meier curves are shown to illustrate survival outcomes within each class by estimating the survival function $S(t)$:

$$S(t) = \prod_{t_i<t} \frac{n_i - d_i}{n_n}, \quad (6)$$

where d_i are the number of recurrence events at time t and n_i are the number of subjects at risk of death or recurrence prior to time t.

3. Results

[0084] Results of model selection by varying λ and J are shown in Table 1. Best performance was measured by CHI with λ set to 0.2 and J set to 5.

[0085] Cox-proportional hazard modeling showed strong significance in recurrence-free survival between patients when classifying their tissue based on clusters produced by the unsupervised model. Table 2 details the hazard ratios of each cluster, relative to the others in five different models. Each model has one cluster held as a reference to produce the hazard ratio. FIG. 8 shows a visualization of the survival model using Kaplan-Meier analysis.

Table 1. The Calinski-Harabaz Index (CHI) was highest when the model was set to 5 clusters (J) and clustering weight (λ) was set to 0.2 This indicates the model which best produces clusters that are dense and well-separated from each other.

J	5	10	15	20	25
CHI	**3863**	2460	2064	957	1261
λ	0.2	0.4	0.6	0.8	1.0
CHI	**4314**	3233	3433	3897	3112

Table 2. Hazard ratios show strong significance as measured by three different statistical tests. This indicates that the cluster classes produced by the unsupervised model suggest

clinical usefulness. If each cluster class is considered a cancer subtype, five subtypes is the strongest stratification seen in the literature thus far.

Cluster	Hazard ratio				
	Dependent variable:				
	0	1	2	3	4
0	Reference	1.332***	0.789***	1.788***	0.873***
		(0.206)	(0.250)	(0.237)	(0.235)
1	0.751***	Reference	0.593***	1.342***	0.655***
	(0.206)		(0.251)	(0.236)	(0.235)
2	1.267***	1.688***	Reference	2.265***	1.106***
	(0.250)	(0.251)		(0.277)	(0.272)
3	0.559***	0.745***	0.441	Reference	0.488*
	(0.237)	(0.236)	(0.277)		(0.264)
4	1.145***	1.526***	0.904***	2.048***	Reference
	(0.235)	(0.235)	(0.272)	(0.264)	
Wald Test	12.740**	12.740**	12.740**	12.740**	12.740**
LR Test	13.183**	13.183**	13.183**	13.183**	13.183**
Logrank Test	13.097**	13.097**	13.097**	13.097**	13.097**

Note: *p<0.1; **p<0.05; ***p<0.01

4. Conclusion

[0086] This model offers a novel approach for identifying histological patterns of potential prognostic significance, circumventing the tasks of tedious tissue labeling and laborious human evaluation of multiple whole slides. As a point of comparison, a recent study showed that an effective prognostic score for colorectal cancer was achieved by first segmenting a slide into eight predefined categorical regions using supervised learning. These kind of models limit the model to pre-defined histologic components (tumor, fat, debris, etc.) and the protocol may not extend to extra-colonic anatomic sites lacking a similar tumor specific stroma interactions. In contrast, the design of the model lacks predefined tissue classes, and has the capability to analyze any number of clusters, thus removing potential bias introduced by training and increasing flexibility in application of the model. Novel subtyping approaches like this may lead to better grading of cholangiocarcinoma and improve treatment and outcome of patients.

C. Systems and Method for Clustering Images Using Auto-Encoder Models

[0087] Referring now to FIG. 9A, depicted is a block diagram of a system 900 for clustering images using autoencoder models. In brief overview, the system 900 may include at least one image classification system 902 (sometimes generally referred to as a computing system), at least one imaging device 904, and at least one display 906, among others. The image classification system 902, the imaging device 904, and the display 906 may be communicatively coupled to one another. The image classification system 902 may include at least one image reconstruction model 908, at least one clustering model 910, at least one image preparer 912, at least one model trainer 914, at least one model applier 916, at least one cluster analyzer 918, and at least one database 920, among others. The at least one database 920 may include or maintain at least one training dataset 922. Each of the components in the system 900 listed above may be implemented using hardware (e.g., one or more processors coupled with memory) or a combination of hardware and software as detailed herein in Section D. Each of the components in the system 900 may implement or execute the functionalities detailed herein, such as those described in Sections A and B.

[0088] In further detail, the image classification system 902 itself and the components therein, such as the image preparer 912, the model trainer 914, the model applier 916, and the cluster analyzer 918, may have a runtime mode (sometimes referred herein as an evaluation mode) and a training mode. Under training mode, the image classification system 902 may train or otherwise update the image reconstruction model 908 and the clustering model 910 using the training dataset 922 maintained on the database 920. Under runtime mode, the image classification system 902 may apply the image reconstruction model 908 and the clustering model 910 using acquired images.

[0089] The image preparer 912 executing on the image classification system 902 may receive, retrieve, or identify at least one image to feed to the image reconstruction model 908. When in training mode, the image preparer 912 may access the database 920 to identify at least one sample image 924A included in the training dataset 922 maintained therein. The training dataset 922 may include one or more sample images 924A. In some embodiments, each sample image 924A of the training dataset 922 may be a biomedical image. The biomedical image may be acquired in accordance with microscopy techniques or a histopathological image preparer, such as using an optical microscope, a confocal microscope, a fluorescence microscope, a phosphorescence microscope, an electron microscope, among others. The biomedical image may be, for example, a histological section with a hematoxylin and eosin (H&E) stain, hemosiderin stain, a Sudan stain, a Schiff stain, a Congo red stain, a Gram stain, a Ziehl-Neelsen stain, a Auramine–rhodamine stain, a trichrome stain, a Silver stain, and Wright's Stain, among others. The biomedical image for the sample image 924A may be from a tissue section from a subject (e.g., human, animal, or plant) for performing histopathological surveys. The tissue sample may be from any part of the subject, such as a muscle tissue, a connective tissue, an epithelial tissue, or a nervous tissue in the case of a human or animal subject. The sample image 924A of the training dataset 922 may be another type of image.

[0090] Each sample image 924A of the training dataset 316 may include one or more regions of interest 926A and 926B (hereinafter generally referred as regions of interest 926). Each region of interest 926 may correspond to areas, sections, or boundaries within the sample image 924A that contain, encompass, or include conditions (e.g., features or objects

within the image). For example, the sample image 924A may be a whole slide image (WSI) for digital pathology of a sample tissue, and the region of interest 926 may correspond to areas with lesions and tumors in the sample tissue. In some embodiments, the regions of interest 926 of the sample image 924A may correspond to different conditions. Each condition may define or specify a classification for the region of interest 926. For example, when the sample image 924A is a WSI of the sample tissue, the conditions may correspond to various histopathological characteristics, such as carcinoma tissue, benign epithelial tissue, stroma tissue, necrotic tissue, and adipose tissue, among others. In the depicted example, the first region of interest 926A may be associated with one condition (e.g., stroma tissue) and the second region of interest 926B may be associated with another condition (e.g., carcinoma tissue).

[0091] Furthermore, each sample image 924A may include or be associated with one or more annotations 928A and 928B (hereinafter generally referred as annotation 928). Each annotation 928 may indicate or label at a portion of one of the regions of interest 926 within the sample image 924A. Each annotation 928 may be at least partially manually prepared by a viewer examining the sample image 924A for conditions. For example, a pathologist examining the biomedical image within the sample image 924A may manually label the regions of interest 926 using the annotations 928 via a computing device. The annotations 928 may indicate, specify, or define an area, dimensions, or coordinates (e.g., using pixel coordinates) of the regions of interest 926 within the sample image 924A.

[0092] In some embodiments, each annotation 928 may identify or indicate different conditions associated with the region of interest 926. For example, when the sample image 924A is a WSI of the sample tissue, the annotation 928 may identify one of the various histopathological characteristics, such as carcinoma tissue, benign epithelial tissue, stroma tissue, necrotic tissue, and adipose tissue, among others. In the depicted example, the first annotation 928A may be associated with one condition (e.g., stroma tissue) corresponding to the first region of interest 926A. Furthermore, the second region of interest 926B may be associated with another condition (e.g., carcinoma tissue) corresponding to the second region of interest 926B.

[0093] In some embodiments, the annotations 928 may fully label or partially label (e.g., as depicted) the corresponding region of interest 926 within the sample image 924A. When fully labeled, the annotation 928 may cover or substantially cover (e.g., 90% or above) the corresponding region of interest 926 within the sample image 924A. Conversely, when partially labeled, the annotation 928 may define a portion of the region of interest 926 (less than fully) within the sample image 924A. In addition, the portion defined by each annotation 928 may be separated from at least one edge of the region of interest 926. The edge may define a perimeter or a boundary between two or more regions of interest 926.

[0094] When in runtime mode, the image preparer 912 may identify at least one input image 924B included in at least one input 930 from the imaging device 904. The imaging device 904 may acquire or generate at least one input image 924B of at least one sample. The sample may include any object or item, the input image 924B of which is acquired via the imaging device 904 (e.g., a camera). The sample may be, for example, a histological section with a hematoxylin and eosin (H&E) stain, hemosiderin stain, a Sudan stain, a Schiff stain, a Congo red stain, a Gram stain, a Ziehl-Neelsen stain, a Auramine-rhodamine stain, a trichrome stain, a Silver stain, and Wright's Stain, among others. The sample may be from a tissue section from a subject (e.g., human, animal, or plant) for performing histopathological surveys. The tissue sample may be from any part of the subject, such as a muscle tissue, a connective tissue, an epithelial tissue, or a nervous tissue in the case of a human or animal subject, among others. The imaging device 904 may acquire the input image 924B of the sample in accordance with microscopy techniques, such as using an optical microscope, a confocal microscope, a fluorescence microscope, a phosphorescence microscope, an electron microscope, among others.

[0095] With the acquisition of the input image 924B, the imaging device 904 may provide, send, or transmit the input 930 including the input image 924B to the image classification system 902. The input image 924B may be similar to the sample image 924A, and may include one or more regions of interest 926 within the input image 924B. On the other hand, the input image 924B may lack any annotations 928 that is associated with the sample image 924A. In some embodiments, the imaging device 904 may acquire multiple input images 924B as a set to provide to the image classification system 902 via the input

930. The image preparer 912 may in turn receive the input 930 including the input image 924B from the imaging device 904.

[0096] Referring now to FIG. 9B, depicted is a block diagram of the image reconstruction model 908 and the clustering model 910 in the context of the system 900. With the identification of the image 924 (generally referring to the sample image 924A or the input image 924B), the image preparer 912 may generate at least one set of tiles 940A–N (hereinafter generally referred as tiles 940). Each tile 940 may correspond to a portion of the image 924. Each tile 940 may have a dimension (or a resolution) as defined for input into the image reconstruction model 908. For example, the dimensions of each tile 940 may be 224 x 224 pixels. In some embodiments, the tile generator 3215 may partition or divide the image 924 into the set of tiles 940. In some embodiments, the image preparer 912 may apply to one or more magnification factors to generate the set of tiles 940. The magnification factors applied to the image 924 may range from 3x to 100x. In some embodiments, the image preparer 912 may generate the set of tiles 940 from the image 924 without overlap. In some embodiments, the image preparer 912 may generate the set of tiles 940 with an overlap of a set ratio. The ratio may range from 10% to 90% overlap between pairs of adjacent tiles 940.

[0097] In some embodiments, the image preparer 912 may identify a portion of the image 924 to generate set of tiles 940. When in training mode, the image preparer 912 may identify the portion of the sample image 924A corresponding to the region of interest 926 from the annotation 928. With the identification of the region of interest 926, the image preparer 912 may use the corresponding portion to generate the set of tiles 940. When in runtime mode, the image preparer 926 may identify the portion of the input image 924B to use in generating the set of tiles 940 by removing negative space. The identification of the negative space may be in accordance with a feature detection algorithm. For example, the negative space region of the input image 924B may lack any portion of the micro-anatomical sample or specimen. The negative space within the input image 924B may correspond to the region of the image 924B that is null or white. The remaining portion of the input image 924B may correspond to the region of interest 926. Using the remaining portion, the image preparer 912 may generate the set of tiles 940.

[0098] In some embodiments, the image preparer 912 may select or identify a subset of tiles 940 from the generated set to feed to the image reconstruction model 908. In some embodiments, the image preparer 912 may first generate the set of tiles 940 from the image 924 and then remove a subset of tiles 940 not corresponding to any regions of interest 926 within the image 924. In some embodiments, the image preparer 912 may perform one or more quality control measures in selecting the subset of tiles 940 to feed. The quality control measures may include noise detection, blur detection, and brightness detection, among others. For example, the image preparer 912 may remove a tile 940 from the set when the tile 940 is determined to have an amount of noise greater than a noise threshold value, an amount of blur greater than a blur tolerance value, or a brightness lower than a brightness threshold value. Conversely, the image preparer 912 may maintain a tile 940 in the set when the tile 940 is determined to have an amount of noise lower than the noise threshold value, an amount of blur lower than the blur tolerance value, or a brightness higher than the brightness threshold value.

[0099] The model trainer 914 executing on the image classification system 902 may initiate, establish, and maintain the image reconstruction model 908. The initiation and establishment of the image reconstruction model 908 may be under training mode and may use the sample images 924A of the training dataset 922 maintained on the database 920. As depicted, the image reconstruction model 908 may include at least one encoder block 942 (sometimes referred herein as a residual network) and at least one decoder block 944 (sometimes referred herein as a decoder network). In some embodiments, the training of the encoder block 942 may be performed separately from the decoder block 944, or vice-versa. For example, the model trainer 914 may using one training dataset 922 to train the encoder block 942. Once the encoder block 942 is trained, the model trainer 914 may train both the encoder block 942 and the decoder block 944 using a different training dataset 922 in the same iteration. In some embodiments, the training of the encoder block 942 and the decoder block 944 may be performed in conjunction. For example, the model trainer 914 may use the same training dataset 922 to train the encoder block 942 and the decoder block 944 in the same epoch or iteration. The image reconstruction model 908 may be an instance of the architecture detailed herein in conjunction with FIGs. 3 and 7 in Sections A and B.

[0100] Each of the encoder block 942 and the decoder block 944 of the image reconstruction model 908 may have at least one input and at least one output. The input to the image reconstruction model 908 may correspond to the input of the encoder block 942. The input may include the set of tiles 940 (or corresponding feature space representations) to be processed one-by-one by encoder block 942. The output of the encoder block 942 may include a corresponding set of embedding representations 940'A–N (hereinafter generally referred to as embedding space 940') corresponding to the set of tiles 940. The set of embedding representations 940' may be collectively referred to as an embedding space. Each embedding representation 940' may be of a lower dimension than the corresponding input tile 940. The output of the encoder block 942 may be feed into the input of the decoder block 944. The input of the decoder block 944 may thus include the embedding space 940'. The output of the decoder block 944 may include a set of reconstructed tiles 940" (or a corresponding feature space representation). Each reconstructed tile 940" may be of a higher dimension than the corresponding embedding representation 940', and may be of the same dimension as the original input tile 940.

[0101] In each of the encoder block 942 and the decoder block 944, the input and the output may be related to each other may include a set of weights to be applied to the input to generate the output. In some embodiments, the set of weights may be arranged in one or more transform layers. Each layer may specify a combination or a sequence of application of the weights. The layers may be arranged in accordance with the machine learning algorithm or model for the image reconstruction model 308, for example, as detailed herein in conjunction with FIGs. 9C–9F.

[0102] The model applier 916 executing on the image classification system 902 may apply the image reconstruction model 908 to the set of tiles 940 generated from the image 924 (e.g., the sample image 924A or the input image 924B). In some embodiments, the set of tiles 940 to be applied may be from multiple images 924. In applying, the model applier 916 may feed each individual tile 940 to the image reconstruction model 908. The model applier 916 may apply the weights of the encoder block 942 to each tile 940 to generate a corresponding embedding representation 940'. The model applier 916 may identify the set of embedding representations 940' outputted by the encoder block 942. The model applier 916

may apply the weights of the decoder block 944 to each embedding representation 940' to generate a corresponding reconstructed tile 940". The model applier 916 may also apply the set of embedding representations 940' to the clustering model 910 to output a corresponding set of classifications 946. Each classification 946 may identify an condition for each tile 940.

[0103] Referring now to FIG. 9C, depicted is a block diagram of the encoder block 942 of the image reconstruction model 908 in the system 900 for clustering images. The encoder block 942 of the image reconstruction model 908 may include at least one input and at least one output. The input can include each tile 940. The output can include an embedding representation 940' (sometimes herein referred as a feature map) corresponding to the tile 940. The encoder block 942 can have a set of convolution stacks 950A–N (hereinafter generally referred to as convolution stacks 950). The input and the output of the encoder block 942 may be related via the weights as defined in the set of convolution stacks 950. The set of convolution stacks 950 can be arranged in series (e.g., as depicted) or parallel configuration, or in any combination. In a series configuration, the input of one convolution stacks 950 may include the output of the previous convolution stacks 950 (e.g., as depicted). In parallel configuration, the input of one convolution stacks 950 may include the input of the entire encoder block 942.

[0104] Referring now to FIG. 9D, depicted is a block diagram of a convolution stack 950 in the encoder block 942 of the image reconstruction model 908 in the system 900. Each convolution stack 950 in the convolution stack 950 can have a set of transform layers 952A–N (hereinafter generally referred to as transform layers 952). The set of transform layers 952 can include one or more weights to modify or otherwise process the input to produce or generate an output feature map 956. The input may include one of the tiles 940 when the convolution stack 950 is the first in the encoder block 942. The input may include a resultant feature map 956 when the convolution stack 950 is not the first in the encoder block 942. The output feature map 956 may correspond to one of the embedding representations 940'. The set of transform layers 952 can be arranged in series, with an output of one transform layer 952 fed as an input to a succeeding transform layer 952. Each transform layer 952 may have a non-linear input-to-output characteristic. In some embodiments, the set of transform layers 952 may be a convolutional neural network (CNN), including a convolutional layer, a

normalization layer, and an activation layer (e.g., a rectified linear unit (ReLU)), among others.

[0105] In some embodiments, at least one of the convolution stacks 950 in the encoder block 942 may include at least one skip aggregator operator 954. The skip aggregator operator 954 can combine the output feature map 956 from the set of transform layers 952 and the original input to generate an output feature map 956 for the convolution stack 950. The combination may include addition, concatenation, or a weighted summation, among others. In some embodiments, the skip aggregator operator 954 can concatenate the output from the set of transform layers 952 with the input into the convolution stack 950 to generate the output feature map 956.

[0106] Referring now to FIG. 9E, depicted is a block diagram of a decoder block 944 of the image reconstruction model 908 in the system 900. The decoder block 944 of the image reconstruction model 908 may include at least one input and at least one output. The input can include one of the embedding representations 940'. The output can include a reconstructed tile 940" corresponding to the input embedding representation 940'. The decoder block 944 can have a set of deconvolution stacks 960A–N (hereinafter generally referred to as deconvolution stacks 960). The input and the output of the decoder block 944 may be related via the weights as defined in the set of deconvolution stacks 960. The set of deconvolution stacks 960 can be arranged in series (e.g., as depicted) or parallel configuration, or in any combination. In a series configuration, the input of one deconvolution stacks 960 may include the output of the previous deconvolution stacks 960 (e.g., as depicted). In parallel configuration, the input of one deconvolution stacks 960 may include the input of the entire decoder block 944.

[0107] Referring now to FIG. 9F, depicted is a block diagram of a deconvolution stack 960 in the decoder block 944 of the image reconstruction model 908 in the system 900. Each deconvolution stack 960 in the deconvolution stack 960 can have at least one up-sampler 964 and a set of transform layers 966A–N (hereinafter generally referred to as transform layers 966). The up-sampler 964 and the set of transform layers 966 can include one or more weights to modify or otherwise process the input to produce or generate an

output reconstructed tile 940". The input may include one of the embedding representations 940' when the deconvolution stack 960 is the first in the decoder block 944. The input may include a resultant feature map 962 from a previous deconvolution stack 960 when the deconvolution stack 960 is not the first in the decoder block 944.

[0108] The up-sampler 964 may increase the image resolution of the feature map 962 to increase a dimension (or resolution) to fit the set of transform layers 966. In some implementations, the up-sampler 964 can apply an up-sampling operation to increase the dimension of the feature map 962. The up-sampling operation may include, for example, expansion and an interpolation filter, among others. In performing the up-sampling operation, the up-sampler 964 may insert null (or default) values into the feature map 962 to expand the dimension. The insertion or null values may separate the pre-existing values. The up-sampler 964 may apply a filter (e.g., a low-pass frequency filter or another smoothing operation) to the expanded feature map. With the application, the up-sampler 964 may feed the resultant feature map 962 into the transform layers 966.

[0109] The set of transform layers 966 can be arranged in series, with an output of one transform layer 966 fed as an input to a succeeding transform layer 966. Each transform layer 966 may have a non-linear input-to-output characteristic. In some embodiments, the set of transform layers 966 may be a convolutional neural network (CNN), including a convolutional layer, a normalization layer, and an activation layer (e.g., a rectified linear unit (ReLU)), among others.

[0110] Referring now to FIG. 9G, depicted is a block diagram of the clustering model 910 in the system 900. The clustering model 910 may include or define at least one feature space 970 (sometimes referred herein as a data space). The feature space 970 may be a *n*-dimensional representation within which each of the embedding representations 940' can reside as a corresponding set of points 972A–N (generally referred herein as points 972). For example, each embedding representation 940' may be 1 x 512 in dimension. In this example, the feature space 970 may be 512-dimensional, and the points 972 may reside somewhere within the feature space 970 based on the values of the corresponding embedding representation 940'.

[0111] The clustering model 910 may include a set of inputs and a set of outputs. The inputs may include the set of embedding representations 940' generated by the encoder block 942 of the image reconstruction model 908. The outputs may include a set of classifications 946 for the set of tiles 940 corresponding to the set of embedding representations 940'. The classification 946 may identify, indicate, or correspond to which condition is present the respective tile 940 (or embedding representation 940'). For example, the classification 946 may indicate that the tile 940 corresponding to the input embedding representation 940' is correlated with a lesion in a tissue sample from which the image 924 is acquired. The inputs and outputs of the clustering model 910 may be related via the feature space 970.

[0112] The feature space 970 of the clustering model 910 may define or include a set of centroids 974A–N (generally referred hereinafter centroids 974) and a set of partitions 976A–N (generally referred herein after as partitions 976). Each centroid 974 may define a corresponding partition 976 within the feature space 970. Each partition 976 may correspond to a portion of the feature space 970 most proximate (e.g., in terms of Euclidean distance or L-norm distance) to the centroid 974 defining the respective partition 976.

[0113] The model trainer 914 may initialize, establish, or maintain the clustering model 910. In initializing, the model trainer 914 may set or assign a set of points within the features space 760 for the corresponding set of centroids 974. The assignment of the points in initializing may occur prior the training of the image reconstruction model 908. The set of points for the centroids 974 may be at random points within the features space 750 (e.g., using a pseudo-random number generator). The set of points for the centroids 974 may be at predefined points within the feature space 970 (e.g., a defined distances from one another). In some embodiments, the model trainer 914 may identify a number of conditions for the regions of interest 926 within the image 924 for the number of centroids 974. In some embodiments, the model trainer 914 may use a predefined number (e.g., as configured by an administrator of the system 900) for the number of centroids 974. Using the number, the model trainer 914 may create the set of points within the feature space 760 for the set of centroids 974.

[0114] Based on the assignment of the set of points for the centroids 974 in the initialization, the model trainer 914 may also identify or determine the set of corresponding partitions 976 within the feature space 970. For each centroid 974, the model trainer 914 may identify the portion of the feature space 970 most proximate to the centroid 974. The model trainer 914 may define the identified portion of the feature space 970 about the centroid 974 as the respective partition 976. In addition, the model trainer 914 may assign or classify each tile 940 using the initial assignment of the set of centroids 974.

[0115] The cluster analyzer 918 executing on the image classification system 902 may apply the set of embedding representations 940' to the clustering model 910 to identify the corresponding set of classifications 946. In applying, the cluster analyzer 918 may place, insert, or otherwise include each of the embedding representations 940' into the feature space 970 defined by the clustering model 910. For each embedding representation 940', the cluster analyzer 918 may determine or identify a corresponding point 972 within the feature space 970. The cluster analyzer 918 may calculate or determine a distance metric (e.g., a Euclidean distance or an L-norm distance) between the identified point 972 and each of the centroids 974. Using the calculated distance metrics, the cluster analyzer 918 may identify the centroid 974 to which the point 972 is most proximate (e.g., in terms of Euclidean distance or L-norm distance).

[0116] With the identification of the centroid 974, the cluster analyzer 918 may determine, assign, or otherwise classify the tile 940 that corresponds to the embedding representation 940' as the condition that corresponds to the centroid 974. In some embodiments, the cluster analyzer 918 may identify the partition 976 under which the point 972 corresponding to the embedding representation 940' resides within the feature space 970. The cluster analyzer 918 may use the partition 976 to identify the condition for the tile 940 that corresponds to the embedding representation 940'. The cluster analyzer 918 may repeat these operations to identify the classifications 946 over the set of embedding representations 940'.

[0117] Based on the classifications, the model trainer 914 may update, adjust, or otherwise modify the feature space 970 of the clustering model 910. The modification of the

features space 970 of the clustering model 910 may be performed under training mode. Using the points 972 within the feature space 970, the model trainer 914 may modify the set of centroids 974 defining the set of partitions 976 for the conditions. For each classification, the model trainer 914 may identify a set of points 972 residing within the partition 976 for the classification. Using the identified set of points 972, the model trainer 914 may calculate or determine a new centroid 974. For example, the model trainer 914 may calculate an average value in each dimension of the feature space 970 as the new centroid 974. Once the new set of centroids 974 are determined, the model trainer 914 may determine or identify a new set of partitions 976 within the feature space 970 in a similar manner as detailed above.

[0118] The model trainer 914 may calculate or determine a clustering error metric also based on the classifications. The clustering error metric may indicate or represent a degree of discrepancy of the tiles 940 to the proper classification within the feature space 970 of the clustering model 910. The clustering error metric may be calculated in accordance with any number of loss functions, such as mean squared error (MSE), a quadratic loss, and a cross-entropy loss, among others. To determine the clustering error metric, the model trainer 914 may identify the set of centroids 974 in the clustering model 910 prior to the modification of the feature space 970. For each centroid 974, the model trainer 914 may identify the set of points 972 that are identified as most proximate to the centroid 974. For each point 972, the model trainer 914 may calculate or determine a distance metric (e.g., Euclidean distance or L-norm distance) between the point and the centroid 974. The calculation of the distance metrics over all the centroids 974 and the points 972 corresponding to the set of embedding representations 940'. Using the distance metrics over all the embedding representations 940', the model trainer 914 may calculate or determine the clustering error metric. For example, the model trainer 914 may calculate an average value or a weighted summation over all the distance metrics as the clustering error metric.

[0119] The model trainer 914 may calculate or determine a reconstruction error metric between the set of tiles 940 and the corresponding set of reconstructed tiles 940". The reconstruction error metric may indicate or represent a degree of deviation between an original tile 940 and a corresponding reconstructed tile 940" generated by the image reconstruction model 908. The reconstruction error metric may be calculated in accordance

with any number of loss functions, such as mean squared error (MSE), a quadratic loss, and a cross-entropy loss, among others. To determine the reconstruction error metric, the model trainer 914 may identify a tile 940 and a reconstructed tile 940" that corresponds to the tile 940. The model trainer 914 may compare the tile 940 with the reconstructed tile 940" to calculate or determine a discrepancy metric. The discrepancy metric may indicate or represent a deviation between the original tile 940 and the reconstructed tile 940". In some embodiments, the model trainer 914 may compare a color value (e.g., red-green-blue or grayscale) of the tile 940 and a color value of the reconstructed tile 940" pixel-by-pixel to determine the discrepancy metric. The calculation or the determination of the discrepancy values may be repeated over each tile 940 or reconstructed tile 940". Using the discrepancy metrics over all the tiles 940 and reconstructed tiles 940", the model trainer 914 may calculate or determine the reconstruction error metric. For example, the model trainer 914 may calculate an average value or a weighted summation over all the discrepancy metrics as the reconstruction error metric.

[0120] In accordance with the clustering error metric and the reconstruction error metric, the model trainer 914 may modify, change, or otherwise update one or more of the weights in the encoder block 942 or the decoder block 944 of image reconstruction model 908. In some embodiments, the model trainer 914 may update the set of transform layers in the image reconstruction model 908. The updating of weights may be in accordance with an optimization function (or an objective function) for the image reconstruction model 908. The optimization function may define one or more rates or parameters at which the weights of the image reconstruction model 908 are to be updated. For example, the model trainer 914 may use the optimization function (e.g., Adam optimization) with a set learning rate (e.g., ranging from 10^{-6} to 10^{-1}), a momentum (e.g., ranging from 0.1 to 1), and a weigh decay (e.g., ranging from 10^{-6} to 10^{-2}) for a number of iterations (e.g., ranging from 10 to 200) in training the image reconstruction model 908.

[0121] In some embodiments, the model trainer 914 may calculate or determine a combined error metric based on the clustering error metric and the reconstruction error metric. The combined error metric may be a weighted summation of the clustering error metric and the reconstruction error metric. In determining the combined error metric, the

model trainer 914 may apply a weight (of factor) to the clustering error metric and a weight (or factor) to the reconstruction error metric. With the combined error metric, the model trainer 914 may update one or more of the weights in the encoder block 942 or the decoder block 944 of the image reconstruction model 908. The updating of the weights in accordance with the optimization function using the combined error metric may be similar as discussed above.

{0122} The model trainer 914 may repeat the training of the image reconstruction model 908 using the sample images 924A of the training dataset 922 maintained on the database 920, and re-apply the tiles 940 to repeat the functionalities as discussed above. In some embodiments, the model trainer 914 may continue training the image reconstruction model 908 by re-applying the tiles 940 for a predefined number of iterations (e.g., as specified by the administrator of the system 900). With each repeated training, the tiles 940 may be repeatedly applied to the image reconstruction model 908. With the re-application of the tiles 940, the classifications 946 at least one of the tiles 940 into the conditions may change relative to the previous assignment. In some embodiments, the model trainer 914 may train the image reconstruction model 908 and the clustering model 910 in accordance with the following pseudo-code:

1. Centroids are randomly initialized in feature space.
2. All samples (tiles) are randomly assigned to a centroid
3. Training begins:
    a. All tiles are pushed thru the model and their embedding (Zi) is saved
    b. Distance between Zi and associated centroid (clustering error) is measured
    c. Reconstruction error between output and input is measured
    d. Combined error is back propagated
    e. Saved Zi is used to update centroid locations by averaging all samples in each cluster
    f. Repeat from step (a). On the repeat Zi will have new values because model was updated in step (d)

[0123] In some embodiments, the model trainer 914 may determine whether to continue training the image reconstruction model 908 based on whether the clustering model 910 is at a convergence state. To determine whether the clustering model 910 is at convergence, the model trainer 914 may identify the set of centroids 974 prior to the modification of the feature space 970. The model trainer 914 may also identify the new set of centroids 974 from the modification of the feature space 970 of the clustering model 910. For each centroid 974 of the previous set, the model trainer 914 may identify a corresponding centroid 974 of the new set. Both centroids 974 may be for the same condition.

[0124] For each pair of centroids 974, the model trainer 914 may calculate or determine a movement metric between the two points corresponding to the centroids 974 within the feature space 970. The movement metric may indicate a degree to which the centroids 974 moved within the feature space 970. The model trainer 914 may also determine a combined movement metric using the movement metrics calculated over each pair of centroids 974. The model trainer 914 may compare the combined movement metric to a threshold value. The threshold value may demarcate a value for the combined movement metric at which the clustering model 910 is deemed to be at convergence.

[0125] When the combined movement metric is less than or equal to the threshold value, the model trainer 914 may determine that the clustering model 910 is at convergence. The model trainer 914 may also terminate the training of the image reconstruction model 908. When the combined movement metric is greater than the threshold value, the model trainer 914 may determine that the clustering model 910 is not at convergence. The model trainer 914 may continue the training of the image reconstruction model 908, and re-apply the tiles 940 from the sample images 924A as discussed above.

[0126] Under runtime mode, the model applier 916 may send, transmit, or provide at least one output 932 for presentation to the display 906. The output 932 may include the image 924 (e.g., sample image 924A or the input image 924B), the set of tiles 940, and the classification 946 for each of the tiles 940. The display 906 may be part of the image classification system 902 or on another computing device that may be communicatively coupled to the image classification system 902. The display 906 may present or render the

output 932 upon receipt from the model applier 916. For example, the display 906 may render a graphical user interface that shows the set of tiles 940 from the image 924. For each tile 940, the graphical user interface rendered on the display 906 may show the condition associated with each of the tiles 940 as determined using the image reconstruction model 908 and the clustering model 910.

[0127] Referring now to FIG. 10A, depicted is a flow diagram of a method 1000 of training autoencoder models to cluster images. The method 1000 may be implemented or performed by any of the components described herein in conjunction with FIGs. 9A–9G or FIG. 11. In overview, a computing system (e.g., the image reconstruction system 902) may identify a sample image (e.g., the sample image 924A) (1005). The computing system may apply an image reconstruction model (e.g., the image reconstruction model 908) to the sample image (1010). The computing system may apply a clustering model (e.g., the clustering model 910) (1015). The computing system may modify a feature space (e.g., the feature space 970) of the clustering model (1020). The computing system may determine a reconstruction error (1025). The computing system may determine a clustering error (1030). The computing system may update weights of the image reconstruction model 908 using the reconstruction error and the clustering error (1035).

[0128] Referring now to FIG. 10B, depicted is a flow diagram of a method 1050 of clustering images using autoencoder models. The method 1050 may be implemented or performed by any of the components described herein in conjunction with FIGs. 9A–9G or FIG. 11. In overview, a computing system (e.g., the image reconstruction system 902) may identify a image (e.g., the input image 924B) (1055). The computing system may apply an image reconstruction model (e.g., the image reconstruction model 908) to the image (1060). The computing system may apply a clustering model (e.g., the clustering model 910) (1065). The computing system may identify a classification from the clustering model (1070).

D. Computing and Network Environment

[0129] Various operations described herein can be implemented on computer systems. FIG. 11 shows a simplified block diagram of a representative server system 1100, client computer system 1114, and network 1126 usable to implement certain embodiments of

the present disclosure. In various embodiments, server system 1100 or similar systems can implement services or servers described herein or portions thereof. Client computer system 1114 or similar systems can implement clients described herein. The system 900 described herein can be similar to the server system 1100. Server system 1100 can have a modular design that incorporates a number of modules 1102 (e.g., blades in a blade server embodiment); while two modules 1102 are shown, any number can be provided. Each module 1102 can include processing unit(s) 1104 and local storage 1106.

[0130] Processing unit(s) 1104 can include a single processor, which can have one or more cores, or multiple processors. In some embodiments, processing unit(s) 1104 can include a general-purpose primary processor as well as one or more special-purpose co-processors such as graphics processors, digital signal processors, or the like. In some embodiments, some or all processing units 1104 can be implemented using customized circuits, such as application specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs). In some embodiments, such integrated circuits execute instructions that are stored on the circuit itself. In other embodiments, processing unit(s) 1104 can execute instructions stored in local storage 1106. Any type of processors in any combination can be included in processing unit(s) 1104.

[0131] Local storage 1106 can include volatile storage media (e.g., DRAM, SRAM, SDRAM, or the like) and/or non-volatile storage media (e.g., magnetic or optical disk, flash memory, or the like). Storage media incorporated in local storage 1106 can be fixed, removable or upgradeable as desired. Local storage 1106 can be physically or logically divided into various subunits such as a system memory, a read-only memory (ROM), and a permanent storage device. The system memory can be a read-and-write memory device or a volatile read-and-write memory, such as dynamic random-access memory. The system memory can store some or all of the instructions and data that processing unit(s) 1104 need at runtime. The ROM can store static data and instructions that are needed by processing unit(s) 1104. The permanent storage device can be a non-volatile read-and-write memory device that can store instructions and data even when module 1102 is powered down. The term "storage medium" as used herein includes any medium in which data can be stored indefinitely (subject to overwriting, electrical disturbance, power loss, or the like) and does

not include carrier waves and transitory electronic signals propagating wirelessly or over wired connections.

[0132] In some embodiments, local storage 1106 can store one or more software programs to be executed by processing unit(s) 1104, such as an operating system and/or programs implementing various server functions such as functions of the system 900 of FIG. 9 or any other system described herein, or any other server(s) associated with system 900 or any other system described herein.

[0133] "Software" refers generally to sequences of instructions that, when executed by processing unit(s) 1104 cause server system 1100 (or portions thereof) to perform various operations, thus defining one or more specific machine embodiments that execute and perform the operations of the software programs. The instructions can be stored as firmware residing in read-only memory and/or program code stored in non-volatile storage media that can be read into volatile working memory for execution by processing unit(s) 1104. Software can be implemented as a single program or a collection of separate programs or program modules that interact as desired. From local storage 1106 (or non-local storage described below), processing unit(s) 1104 can retrieve program instructions to execute and data to process in order to execute various operations described above.

[0134] In some server systems 1100, multiple modules 1102 can be interconnected via a bus or other interconnect 1108, forming a local area network that supports communication between modules 1102 and other components of server system 1100. Interconnect 1108 can be implemented using various technologies including server racks, hubs, routers, etc.

[0135] A wide area network (WAN) interface 1110 can provide data communication capability between the local area network (interconnect 1108) and the network 1126, such as the Internet. Technologies can be used, including wired (e.g., Ethernet, IEEE 802.3 standards) and/or wireless technologies (e.g., Wi-Fi, IEEE 802.11 standards).

[0136] In some embodiments, local storage 1106 is intended to provide working memory for processing unit(s) 1104, providing fast access to programs and/or data to be

processed while reducing traffic on interconnect 1108. Storage for larger quantities of data can be provided on the local area network by one or more mass storage subsystems 1112 that can be connected to interconnect 1108. Mass storage subsystem 1112 can be based on magnetic, optical, semiconductor, or other data storage media. Direct attached storage, storage area networks, network-attached storage, and the like can be used. Any data stores or other collections of data described herein as being produced, consumed, or maintained by a service or server can be stored in mass storage subsystem 1112. In some embodiments, additional data storage resources may be accessible via WAN interface 1110 (potentially with increased latency).

[0137] Server system 1100 can operate in response to requests received via WAN interface 1110. For example, one of modules 1102 can implement a supervisory function and assign discrete tasks to other modules 1102 in response to received requests. Work allocation techniques can be used. As requests are processed, results can be returned to the requester via WAN interface 1110. Such operation can generally be automated. Further, in some embodiments, WAN interface 1110 can connect multiple server systems 1100 to each other, providing scalable systems capable of managing high volumes of activity. Other techniques for managing server systems and server farms (collections of server systems that cooperate) can be used, including dynamic resource allocation and reallocation.

[0138] Server system 1100 can interact with various user-owned or user-operated devices via a wide-area network such as the Internet. An example of a user-operated device is shown in FIG. 11 as client computing system 1114. Client computing system 1114 can be implemented, for example, as a consumer device such as a smartphone, other mobile phone, tablet computer, wearable computing device (e.g., smart watch, eyeglasses), desktop computer, laptop computer, and so on.

[0139] For example, client computing system 1114 can communicate via WAN interface 1110. Client computing system 1114 can include computer components such as processing unit(s) 1116, storage device 1118, network interface 1120, user input device 1122, and user output device 1124. Client computing system 1114 can be a computing device implemented in a variety of form factors, such as a desktop computer, laptop computer, tablet

computer, smartphone, other mobile computing device, wearable computing device, or the like.

[0140] Processor 1116 and storage device 1118 can be similar to processing unit(s) 1104 and local storage 1106 described above. Suitable devices can be selected based on the demands to be placed on client computing system 1114; for example, client computing system 1114 can be implemented as a "thin" client with limited processing capability or as a high-powered computing device. Client computing system 1114 can be provisioned with program code executable by processing unit(s) 1116 to enable various interactions with server system 1100.

[0141] Network interface 1120 can provide a connection to the network 1126, such as a wide area network (e.g., the Internet) to which WAN interface 1110 of server system 1100 is also connected. In various embodiments, network interface 1120 can include a wired interface (e.g., Ethernet) and/or a wireless interface implementing various RF data communication standards such as Wi-Fi, Bluetooth, or cellular data network standards (e.g., 3G, 4G, LTE, etc.).

[0142] User input device 1122 can include any device (or devices) via which a user can provide signals to client computing system 1114; client computing system 1114 can interpret the signals as indicative of particular user requests or information. In various embodiments, user input device 1122 can include any or all of a keyboard, touch pad, touch screen, mouse or other pointing device, scroll wheel, click wheel, dial, button, switch, keypad, microphone, and so on.

[0143] User output device 1124 can include any device via which client computing system 1114 can provide information to a user. For example, user output device 1124 can include a display to display images generated by or delivered to client computing system 1114. The display can incorporate various image generation technologies, e.g., a liquid crystal display (LCD), light-emitting diode (LED) including organic light-emitting diodes (OLED), projection system, cathode ray tube (CRT), or the like, together with supporting electronics (e.g., digital-to-analog or analog-to-digital converters, signal processors, or the like). Some embodiments can include a device such as a touchscreen that function as both

input and output device. In some embodiments, other user output devices 1124 can be provided in addition to or instead of a display. Examples include indicator lights, speakers, tactile "display" devices, printers, and so on.

[0144] Some embodiments include electronic components, such as microprocessors, storage and memory that store computer program instructions in a computer readable storage medium. Many of the features described in this specification can be implemented as processes that are specified as a set of program instructions encoded on a computer readable storage medium. When these program instructions are executed by one or more processing units, they cause the processing unit(s) to perform various operation indicated in the program instructions. Examples of program instructions or computer code include machine code, such as is produced by a compiler, and files including higher-level code that are executed by a computer, an electronic component, or a microprocessor using an interpreter. Through suitable programming, processing unit(s) 1104 and 1116 can provide various functionality for server system 1100 and client computing system 1114, including any of the functionality described herein as being performed by a server or client, or other functionality.

[0145] It will be appreciated that server system 1100 and client computing system 1114 are illustrative and that variations and modifications are possible. Computer systems used in connection with embodiments of the present disclosure can have other capabilities not specifically described here. Further, while server system 1100 and client computing system 1114 are described with reference to particular blocks, it is to be understood that these blocks are defined for convenience of description and are not intended to imply a particular physical arrangement of component parts. For instance, different blocks can be but need not be located in the same facility, in the same server rack, or on the same motherboard. Further, the blocks need not correspond to physically distinct components. Blocks can be configured to perform various operations, e.g., by programming a processor or providing appropriate control circuitry, and various blocks might or might not be reconfigurable depending on how the initial configuration is obtained. Embodiments of the present disclosure can be realized in a variety of apparatus including electronic devices implemented using any combination of circuitry and software.

[0146] While the disclosure has been described with respect to specific embodiments, one skilled in the art will recognize that numerous modifications are possible. Embodiments of the disclosure can be realized using a variety of computer systems and communication technologies including but not limited to specific examples described herein. Embodiments of the present disclosure can be realized using any combination of dedicated components and/or programmable processors and/or other programmable devices. The various processes described herein can be implemented on the same processor or different processors in any combination. Where components are described as being configured to perform certain operations, such configuration can be accomplished, e.g., by designing electronic circuits to perform the operation, by programming programmable electronic circuits (such as microprocessors) to perform the operation, or any combination thereof. Further, while the embodiments described above may make reference to specific hardware and software components, those skilled in the art will appreciate that different combinations of hardware and/or software components may also be used and that particular operations described as being implemented in hardware might also be implemented in software or vice versa.

[0147] Computer programs incorporating various features of the present disclosure may be encoded and stored on various computer readable storage media; suitable media include magnetic disk or tape, optical storage media such as compact disk (CD) or DVD (digital versatile disk), flash memory, and other non-transitory media. Computer readable media encoded with the program code may be packaged with a compatible electronic device, or the program code may be provided separately from electronic devices (e.g., via Internet download or as a separately packaged computer-readable storage medium).

[0148] Thus, although the disclosure has been described with respect to specific embodiments, it will be appreciated that the disclosure is intended to cover all modifications and equivalents within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method of training encoder-decoder models to cluster images, comprising:

identifying, by a computing system having one or more processors, a plurality of tiles derived from a sample image of a training dataset, each of the plurality of tiles corresponding to one of a plurality of conditions and having a first dimension;

applying, by the computing system, an image reconstruction model to the plurality of tiles, the image reconstruction model comprising:

an encoder block having a first set of weights to generate a plurality of embedding representations corresponding to the plurality of tiles, each of the plurality of embedding representations having a second dimension lower than the first dimension; and

a decoder block having a second set of weights to generate a plurality of reconstructed tiles corresponding to the plurality of embedding representations, each of the plurality of reconstructed tiles having a third dimension higher than the second dimension;

applying, by the computing system, a clustering model comprising a feature space to the plurality of embedding representations to classify each the corresponding plurality of tiles to one of the plurality of conditions;

modifying, by the computing system, the feature space of the clustering model based on classifying of the plurality of embedding representations to one of the plurality of conditions;

determining, by the computing system, a first error metric between the plurality of tiles and the corresponding plurality of reconstructed tiles;

determining, by the computing system, a second error metric based on classifying of the plurality of embedding representations to one of the plurality of conditions; and

updating, by the computing system, at least one weight of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance with the first error metric and the second error metric.

2. The method of claim 1, further comprising:

identifying, by the computing system, a plurality of centroids defined by the feature space of the clustering model prior to applying of the clustering model to the plurality of

embedding representations, the plurality of centroids corresponding to the plurality of conditions; and

identifying, by the computing system, a plurality of points defined within the feature space of the clustering model for the corresponding plurality of embedding representations, and

wherein determining the second error metric further comprises determining the second error metric between the plurality of centroids and the plurality of points.

3. The method of claim 1, further comprising determining, by the computing system, a combined error metric in accordance with a weighted summation of the first error metric and the second error metric; and

wherein updating the at least one weight further comprises updating the at least one of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance to the combined error metric.

4. The method of claim 1, further comprising:

determining, by the computing system, that the clustering model is not at a convergence state based on a comparison of a movement metric for a plurality of centroids in the feature space to a threshold value; and

reapplying, by the computing system, the image reconstruction model to the plurality of tiles responsive to determining that clustering model is not at the convergence state.

5. The method of claim 1, further comprising:

applying, by the computing system, subsequent to updating the at least one weight, the image reconstruction to the plurality of tiles, the encoder block to generate a second plurality of embedding representations corresponding to the plurality of tiles; and

applying, by the computing system, the clustering model to the second plurality of embedding representations to classify at least one of the plurality of tiles to a first condition of the plurality of conditions different from a second condition of the plurality of conditions as classified prior to the modifying of the feature space.

6. The method of claim 1, further comprising initializing, by the computing system, the clustering model comprising the feature space to define a plurality of centroids corresponding to the plurality of conditions, each of the plurality of centroids at least one of a random point or a predefined point within feature space.

7. The method of claim 1, wherein identifying the plurality of tiles further comprises:

identifying, from the sample image, a region of interest corresponding to one of the plurality of conditions labeled by an annotation of the training dataset for the sample image; and

generating, using the region of interest identified from the sample image, the plurality of tiles from the sample image.

8. The method of claim 1, wherein identifying the plurality of tiles further comprises identifying the plurality of tiles derived from the sample image of the training dataset, the sample image derived from a tissue sample via a histopathological image preparer, the sample image including a region of interest corresponding to one of the plurality of conditions present in the tissue sample.

9. A method of clustering images using encoder-decoder models, comprising:

identifying, by a computing system having one or more processors, a plurality of tiles derived from an image acquired via an image acquisition device, each of the plurality of tiles having a first dimension;

applying, by the computing system, an image reconstruction model to the plurality of tiles, the image reconstruction model comprising:

an encoder block having a first set of weights to generate a plurality of embedding representations corresponding to the plurality of tiles, each of the plurality of embedding representations having a second dimension lower than the first dimension; and

a decoder block having a second set of weights to generate a plurality of reconstructed tiles corresponding to the plurality of embedding representations; and

applying, by the computing system, a clustering model comprising a feature space to the plurality of embedding representations to classify each the corresponding plurality of tiles to one of a plurality of conditions.

10. The method of claim 9, wherein applying the clustering model further comprises the applying clustering model comprising the feature space, the feature space defining a plurality of centroids corresponding to the plurality of conditions to classify each of the corresponding plurality of tiles to one of the plurality of conditions.

11. The method of claim 9, wherein identifying the plurality of tiles further comprises:

identifying, from the image, a region of interest corresponding to one of the plurality of conditions; and

generating, using the region of interest identified from the image, the plurality of tiles from the image.

12. The method of claim 9, wherein identifying the plurality of tiles further comprises identifying the plurality of tiles derived from the image, the image derived from a tissue sample via a histopathological image preparer, the image including a region of interest corresponding to one of the plurality of conditions present in the tissue sample.

13. The method of claim 9, further comprising training, by the computing system, the image reconstruction model to modify at least one weight of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance with an error metric determined using the clustering algorithm.

14. A system for training encoder-decoder models to cluster images, comprising:

a computing system having one or more processors coupled with memory, configured to:

identify a plurality of tiles derived from a sample image of a training dataset, each of the plurality of tiles corresponding to one of a plurality of conditions and having a first dimension;

apply an image reconstruction model to the plurality of tiles, the image reconstruction model comprising:

an encoder block having a first set of weights to generate a plurality of embedding representations corresponding to the plurality of tiles, each of the plurality of embedding representations having a second dimension lower than the first dimension; and

a decoder block having a second set of weights to generate a plurality of reconstructed tiles corresponding to the plurality of embedding representations, each of the plurality of reconstructed tiles having a third dimension higher than the second dimension;

apply a clustering model comprising a feature space to the plurality of embedding representations to classify each the corresponding plurality of tiles to one of the plurality of conditions;

modify the feature space of the clustering model based on classifying of the plurality of embedding representations to one of the plurality of conditions;

determine a first error metric between the plurality of tiles and the corresponding plurality of reconstructed tiles;

determine a second error metric based on classifying of the plurality of embedding representations to one of the plurality of conditions; and

update at least one weight of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance with the first error metric and the second error metric.

15. The system of claim 14, wherein the computing system is further configured to:

identify a plurality of centroids defined by the feature space of the clustering model prior to applying of the clustering model to the plurality of embedding representations, the plurality of centroids corresponding to the plurality of conditions; and

identify a plurality of points defined within the feature space of the clustering model for the corresponding plurality of embedding representations, and

determine the second error metric between the plurality of centroids and the plurality of points.

16. The system of claim 14, wherein the computing system is further configured to:

determine a combined error metric in accordance with a weighted summation of the first error metric and the second error metric; and

update the at least one of (i) the first set of weights of the encoder block or (ii) the second set of weights of the decoder block in accordance to the combined error metric.

17. The system of claim 14, wherein the computing system is further configured to:

determine that the clustering model is not at a convergence state based on a comparison of a movement metric for a plurality of centroids in the feature space to a threshold value; and

reapply the image reconstruction model to the plurality of tiles responsive to determining that clustering model is not at the convergence state.

18. The system of claim 14, wherein the computing system is further configured to:

apply, subsequent to updating the at least one weight, the image reconstruction to the plurality of tiles, the encoder block to generate a second plurality of embedding representations corresponding to the plurality of tiles; and

apply the clustering model to the second plurality of embedding representations to classify at least one of the plurality of tiles to a first condition of the plurality of conditions different from a second condition of the plurality of conditions as classified prior to the modifying of the feature space.

19. The system of claim 14, wherein the computing system is further configured to:

identify, from the sample image, a region of interest corresponding to one of the plurality of conditions labeled by an annotation of the training dataset for the sample image; and

generate, using the region of interest identified from the sample image, the plurality of tiles from the sample image.

20. The system of claim 14, wherein the computing system is further configured to identify the plurality of tiles derived from the sample image of the training dataset, the sample image

derived from a tissue sample via a histopathological image preparer, the sample image including a region of interest corresponding to one of the plurality of conditions present in the tissue sample.

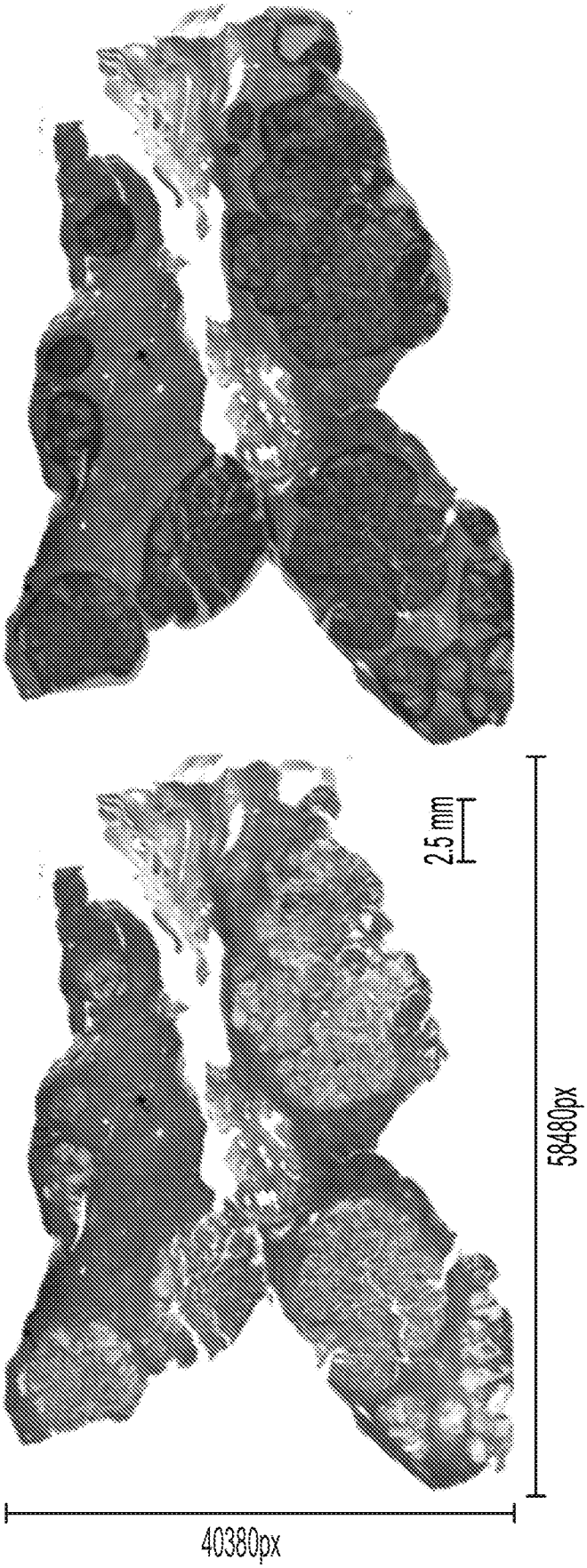

FIG. 1

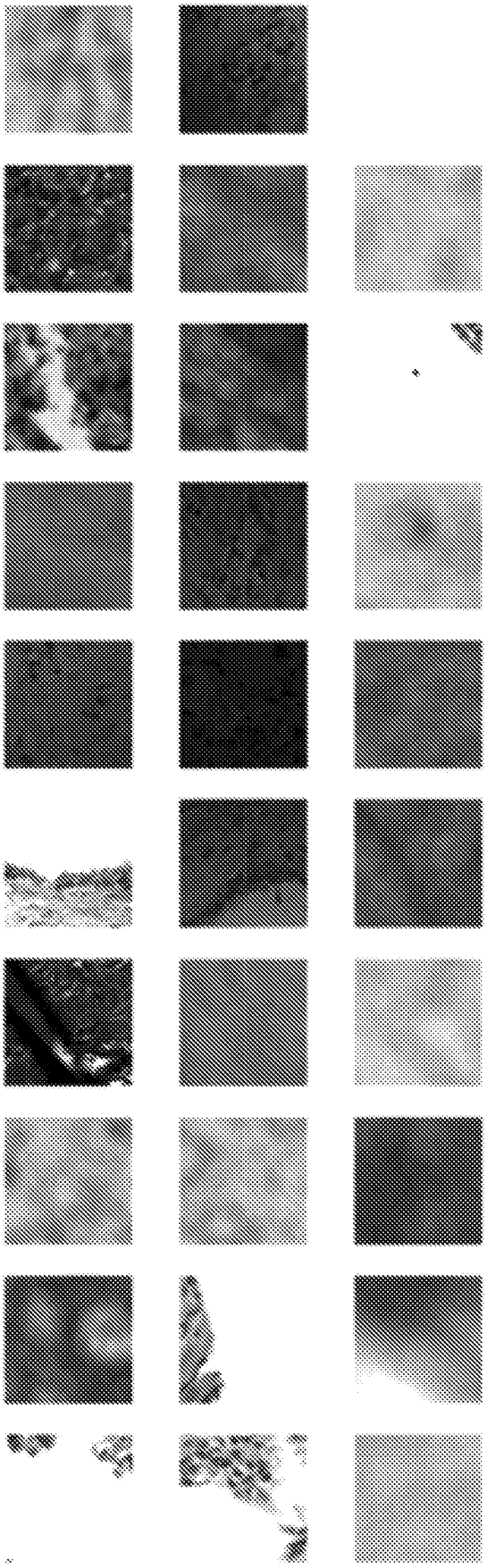

FIG. 2

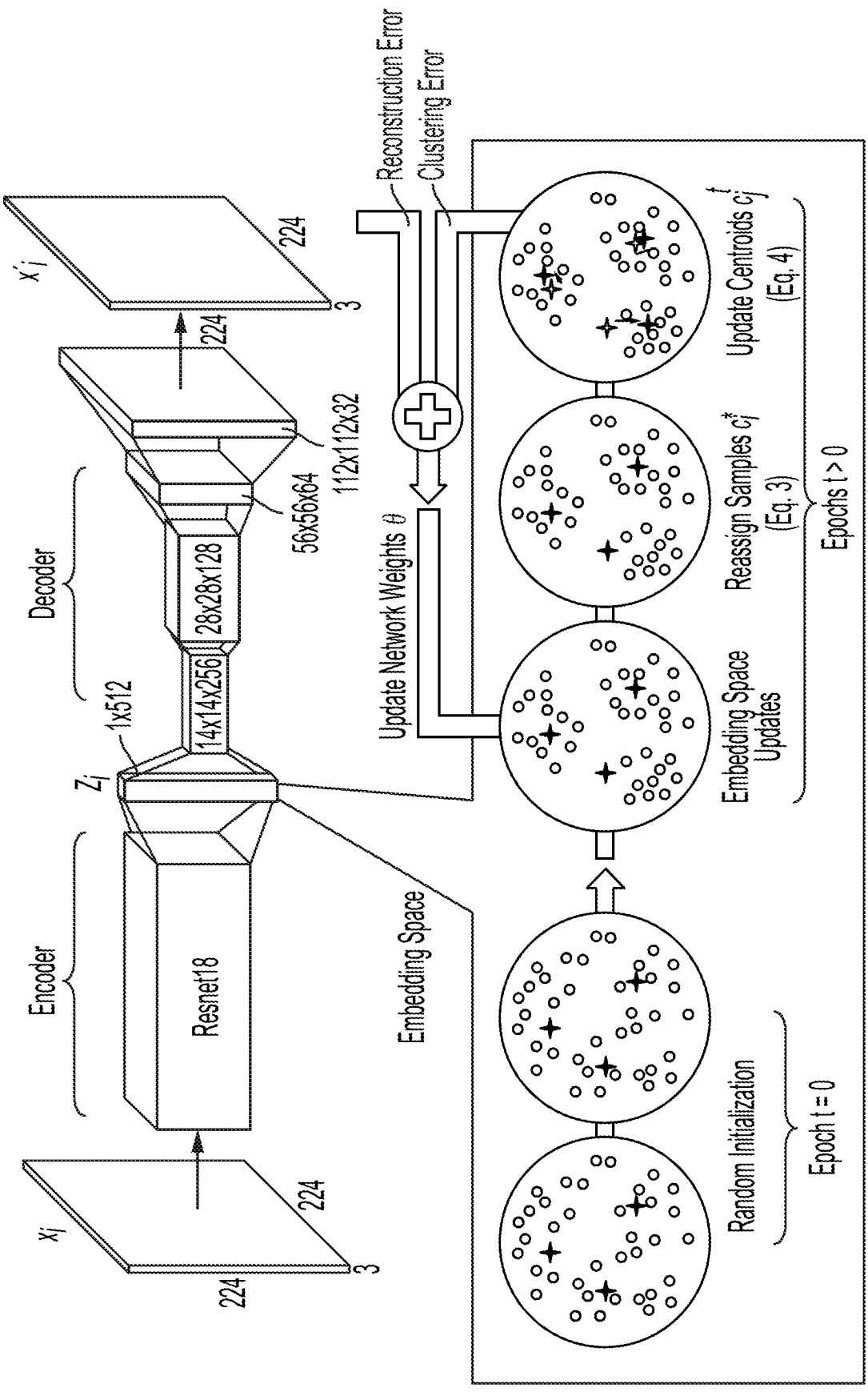
Reconstruction Error
Clustering Error
Update Network Weights θ
Embedding Space
Encoder
Resnet18
Decoder
Zi
1x512
14x14x256
28x28x128
56x56x64
112x112x32
xi
x'i
224
3
Random Initialization
Epoch t = 0
Embedding Space Updates
Reassign Samples c_j^t (Eq. 3)
Update Centroids c_j^t (Eq. 4)
Epochs t > 0

FIG. 3

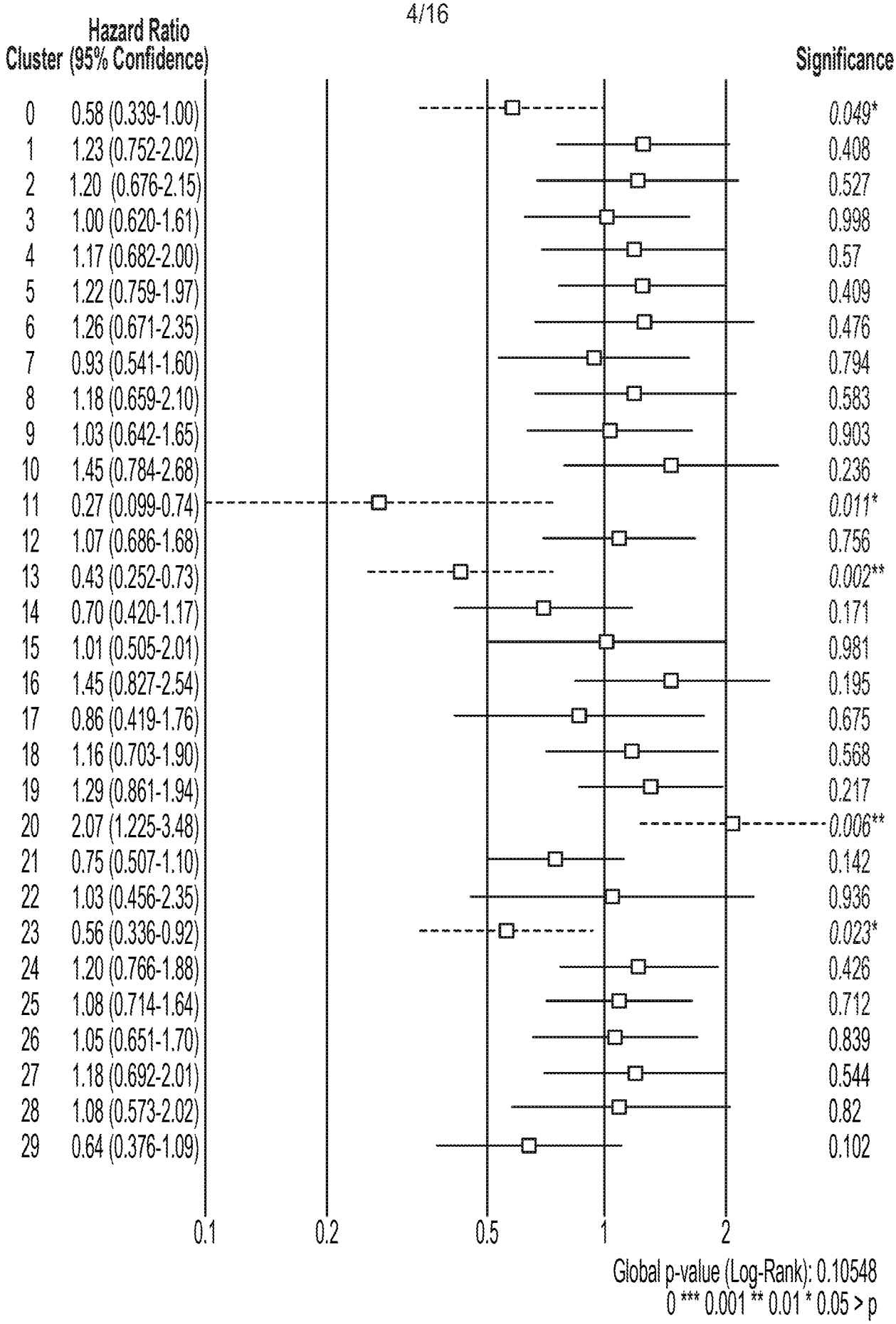
Cluster
Hazard Ratio (95% Confidence)
Significance
0 0.58 (0.339-1.00) 0.049*
1 1.23 (0.752-2.02) 0.408
2 1.20 (0.676-2.15) 0.527
3 1.00 (0.620-1.61) 0.998
4 1.17 (0.682-2.00) 0.57
5 1.22 (0.759-1.97) 0.409
6 1.26 (0.671-2.35) 0.476
7 0.93 (0.541-1.60) 0.794
8 1.18 (0.659-2.10) 0.583
9 1.03 (0.642-1.65) 0.903
10 1.45 (0.784-2.68) 0.236
11 0.27 (0.099-0.74) 0.011*
12 1.07 (0.686-1.68) 0.756
13 0.43 (0.252-0.73) 0.002**
14 0.70 (0.420-1.17) 0.171
15 1.01 (0.505-2.01) 0.981
16 1.45 (0.827-2.54) 0.195
17 0.86 (0.419-1.76) 0.675
18 1.16 (0.703-1.90) 0.568
19 1.29 (0.861-1.94) 0.217
20 2.07 (1.225-3.48) 0.006**
21 0.75 (0.507-1.10) 0.142
22 1.03 (0.456-2.35) 0.936
23 0.56 (0.336-0.92) 0.023*
24 1.20 (0.766-1.88) 0.426
25 1.08 (0.714-1.64) 0.712
26 1.05 (0.651-1.70) 0.839
27 1.18 (0.692-2.01) 0.544
28 1.08 (0.573-2.02) 0.82
29 0.64 (0.376-1.09) 0.102
0.1
0.2
0.5
1
2
Global p-value (Log-Rank): 0.10548
0 *** 0.001 ** 0.01 * 0.05 > p

FIG. 4

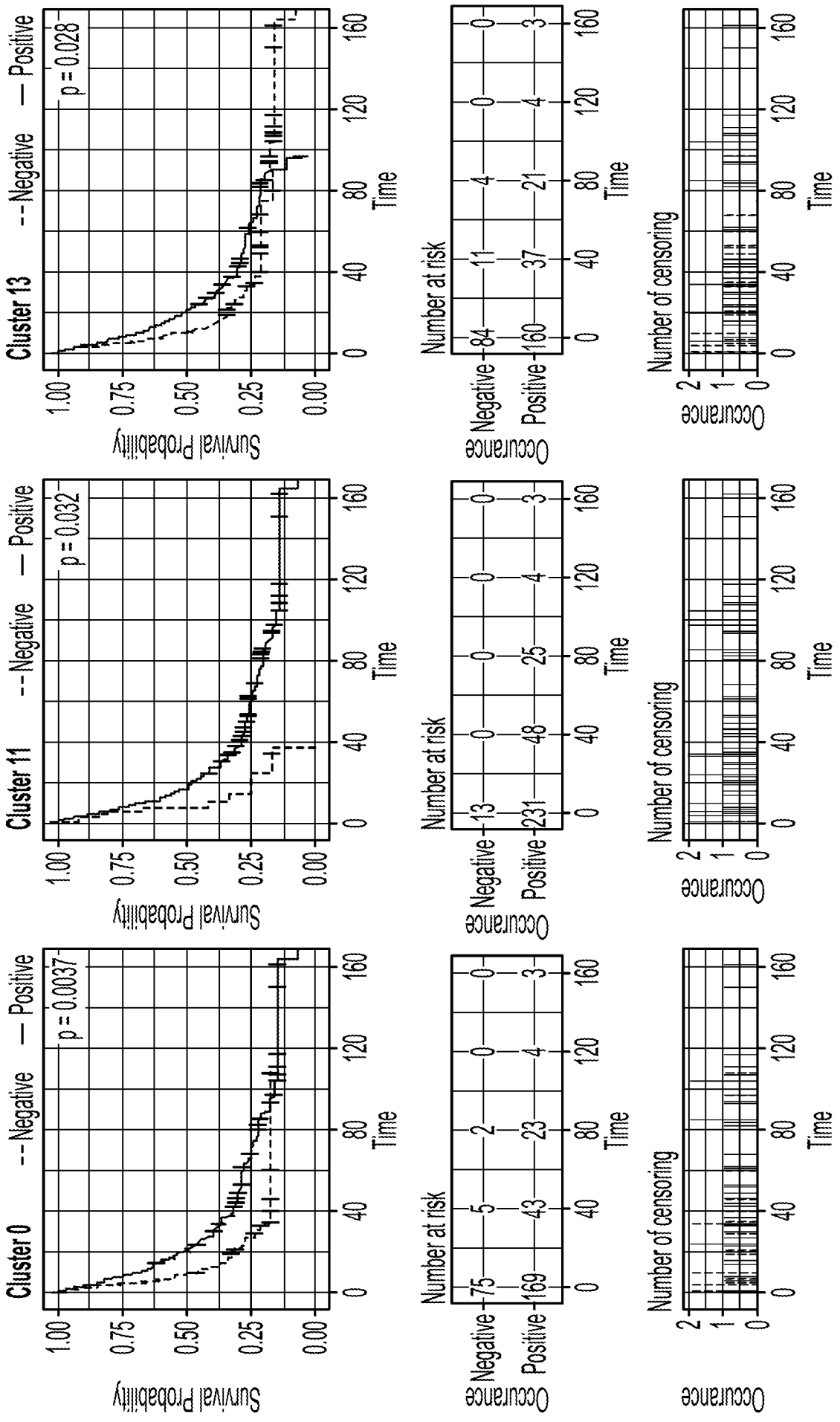
Cluster 13
--Negative — Positive
p = 0.028
Survival Probability
Time
Number at risk
Negative 84 11 4 0 0
Positive 160 37 21 4 3
Occurance
Number of censoring
Cluster 11
--Negative — Positive
p = 0.032
Survival Probability
Time
Number at risk
Negative 13 0 0 0 0
Positive 231 48 25 4 3
Occurance
Number of censoring
Cluster 0
--Negative — Positive
p = 0.0037
Survival Probability
Time
Number at risk
Negative 75 5 2 0 0
Positive 169 43 23 4 3
Occurance
Number of censoring

FIG. 5

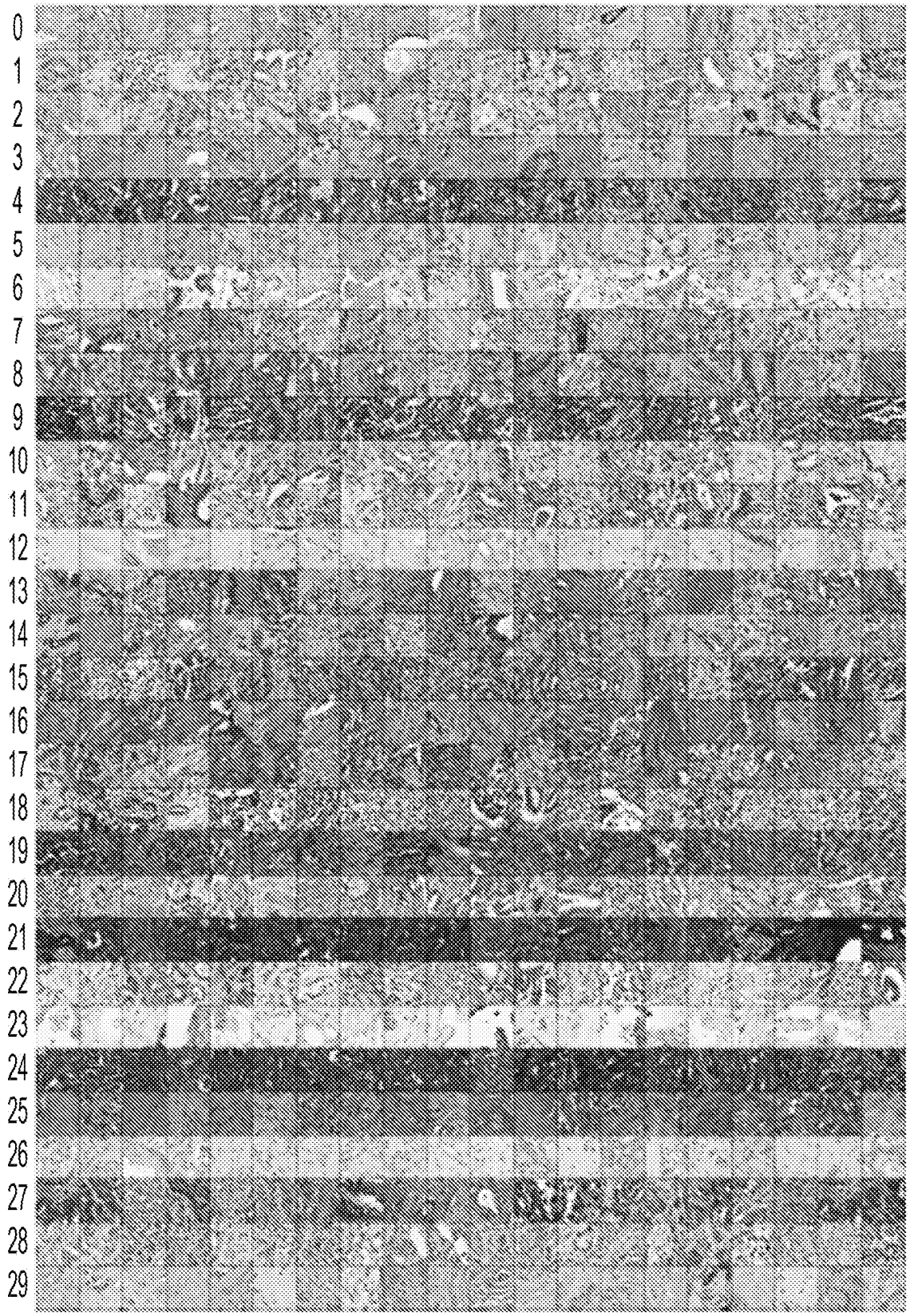
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29

FIG. 6

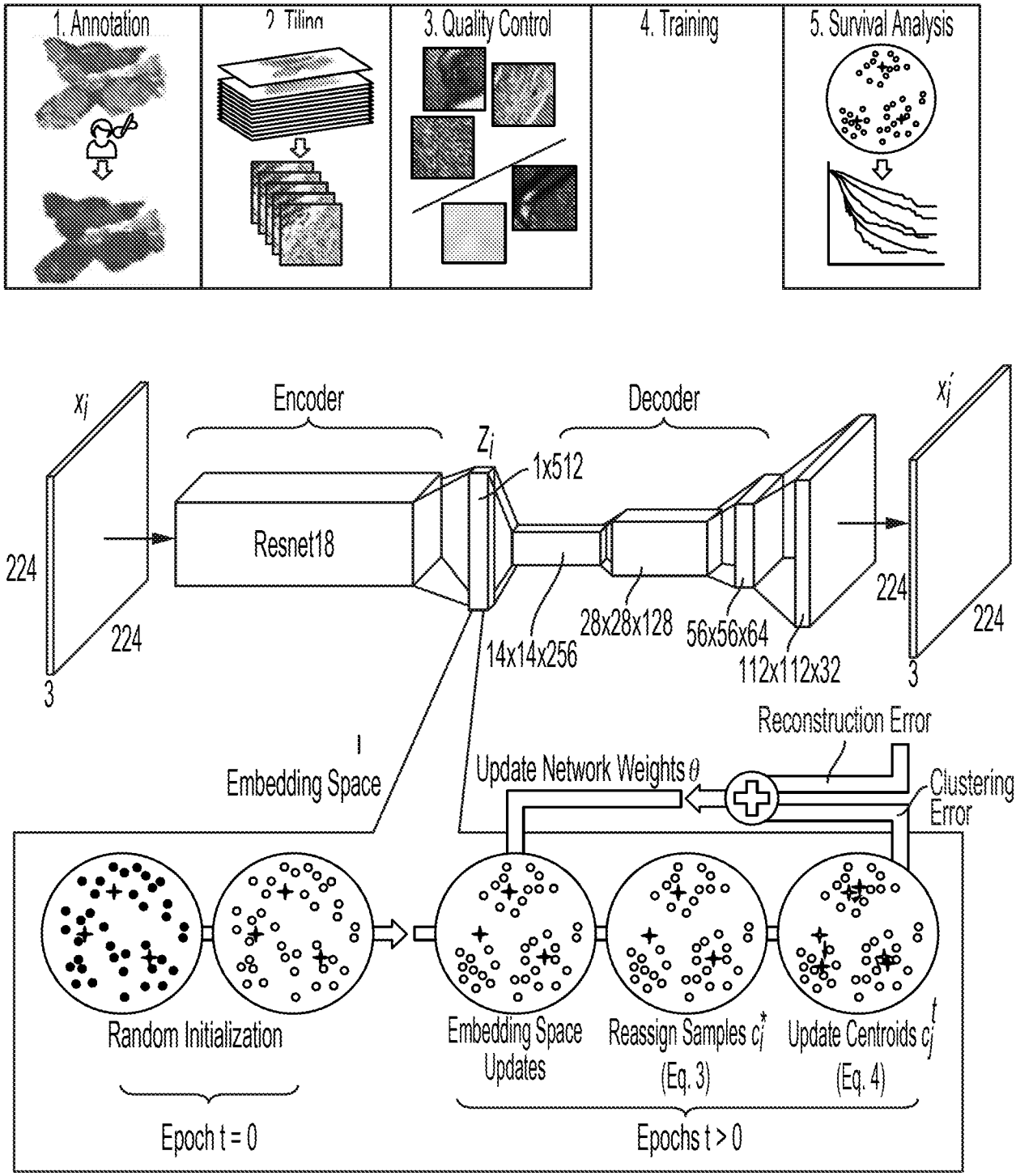
1. Annotation
2. Tiling
3. Quality Control
4. Training
5. Survival Analysis
x_i
Encoder
Decoder
z_i
1x512
Resnet18
224
224
3
14x14x256
28x28x128
56x56x64
112x112x32
x_i'
224
224
3
Reconstruction Error
Update Network Weights θ
Clustering Error
Embedding Space
Random Initialization
Embedding Space Updates
Reassign Samples c_j^t (Eq. 3)
Update Centroids c_j^t (Eq. 4)
Epoch t = 0
Epochs t > 0

FIG. 7

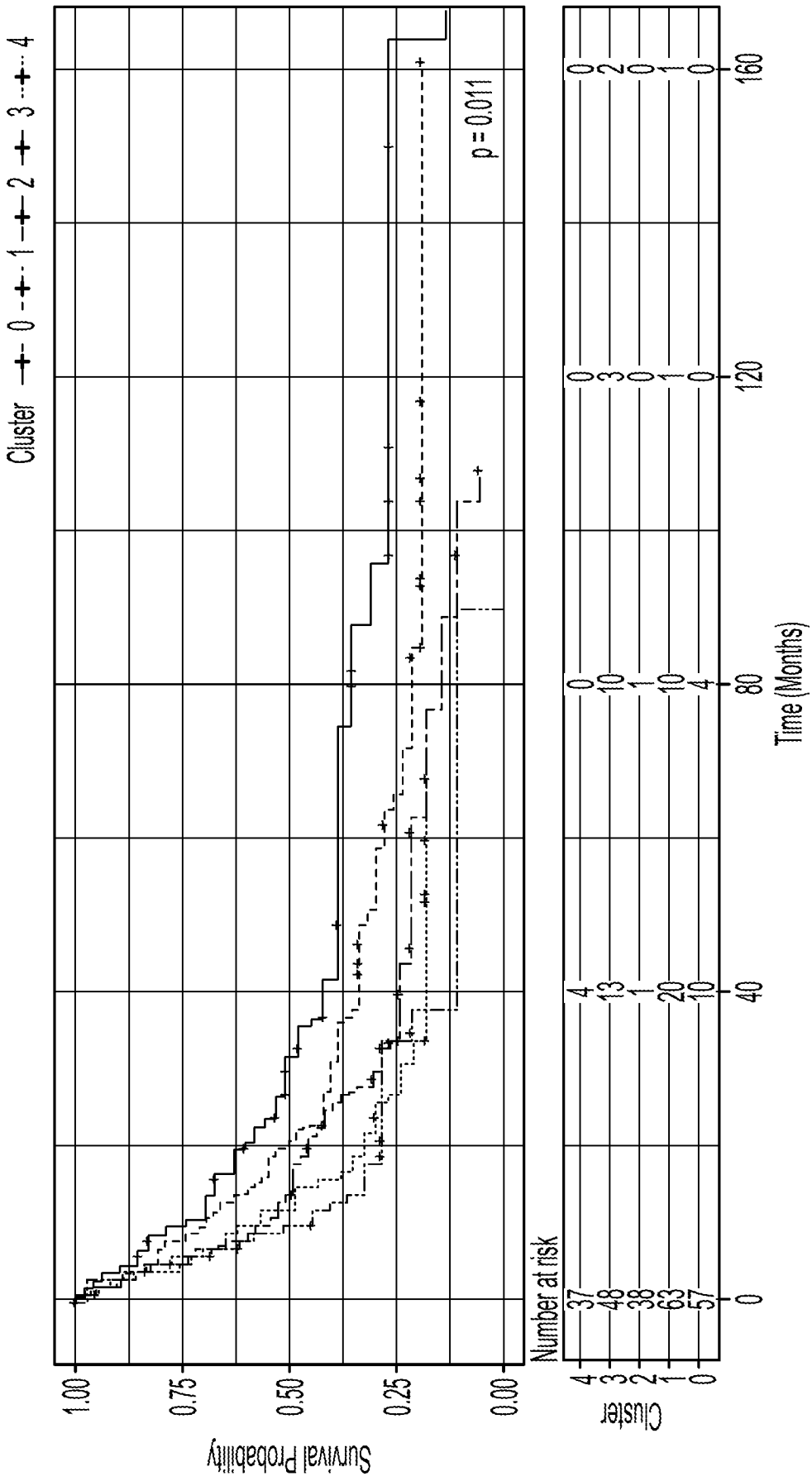
Cluster 0 1 2 3 4
p = 0.011
Survival Probability
1.00
0.75
0.50
0.25
0.00
Number at risk
Cluster
4 37 4 0 0 0
3 48 13 2 3 2 0
2 38 1 0 1 0 0
1 63 20 10 10 4 1 0
0 57 10 0 4 0 0 0
0 40 80 120 160
Time (Months)

FIG. 8

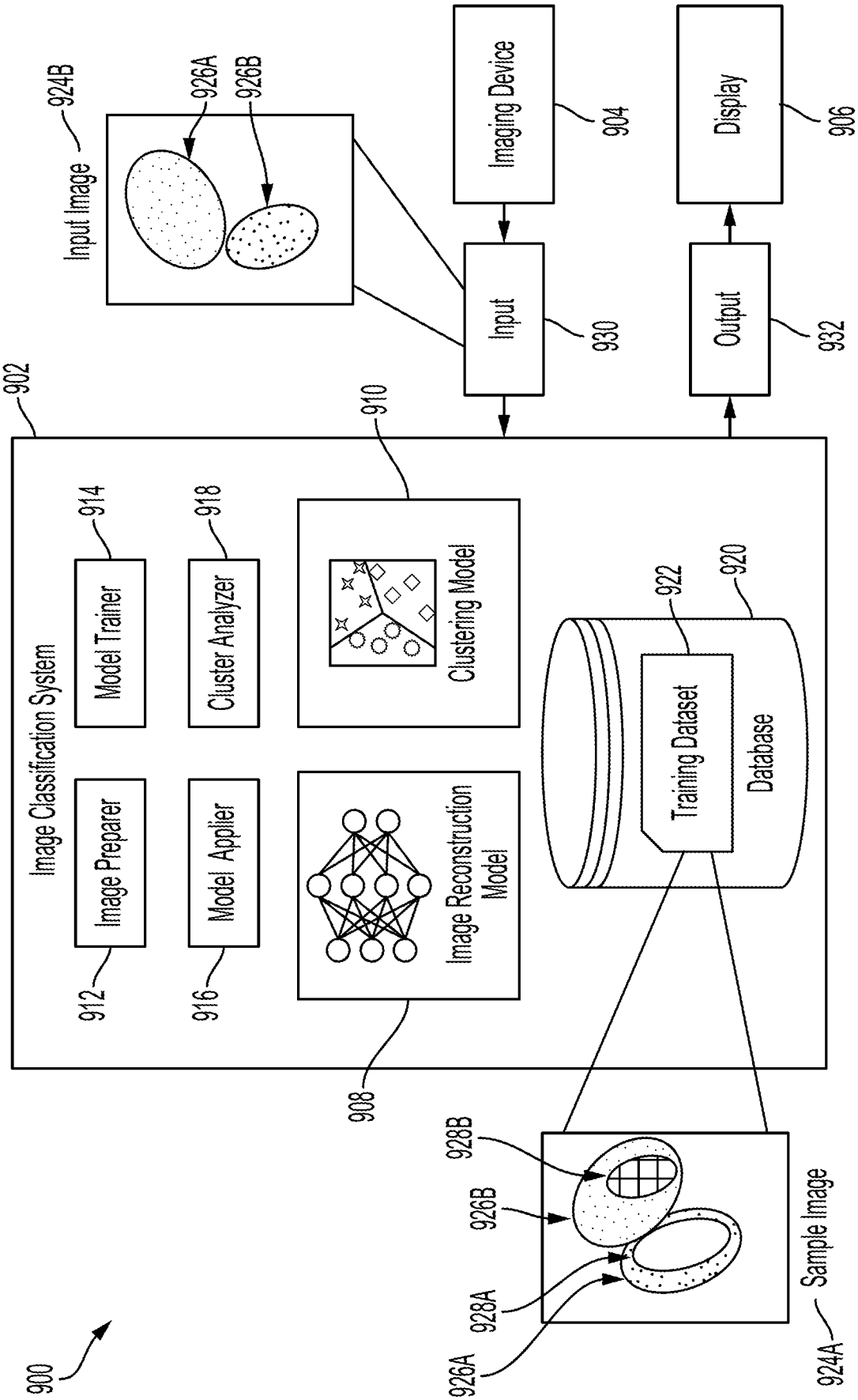
900
902
Image Classification System
Model Trainer
914
Cluster Analyzer
918
Clustering Model
910
Image Preparer
912
Model Applier
916
Image Reconstruction Model
908
Database
920
Training Dataset
922
Input Image
924B
926A
926B
Imaging Device
904
Display
906
Input
930
Output
932
Sample Image
924A
928B
926B
928A
926A

FIG. 9A

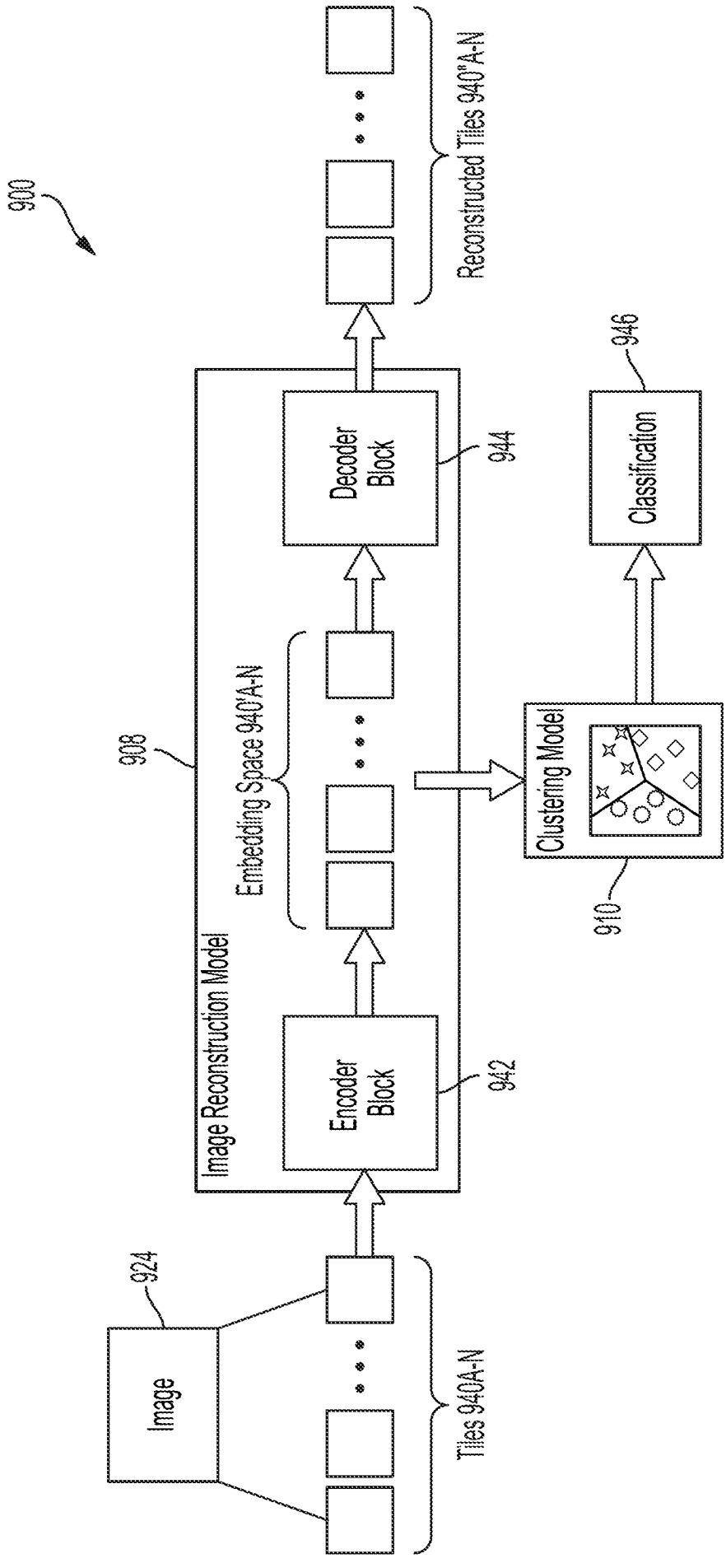
900
Reconstructed Tiles 940'A-N
946
Classification
Decoder Block
944
908
Embedding Space 940A-N
Clustering Model
Image Reconstruction Model
910
Encoder Block
942
924
Image
Tiles 940A-N

FIG. 9B

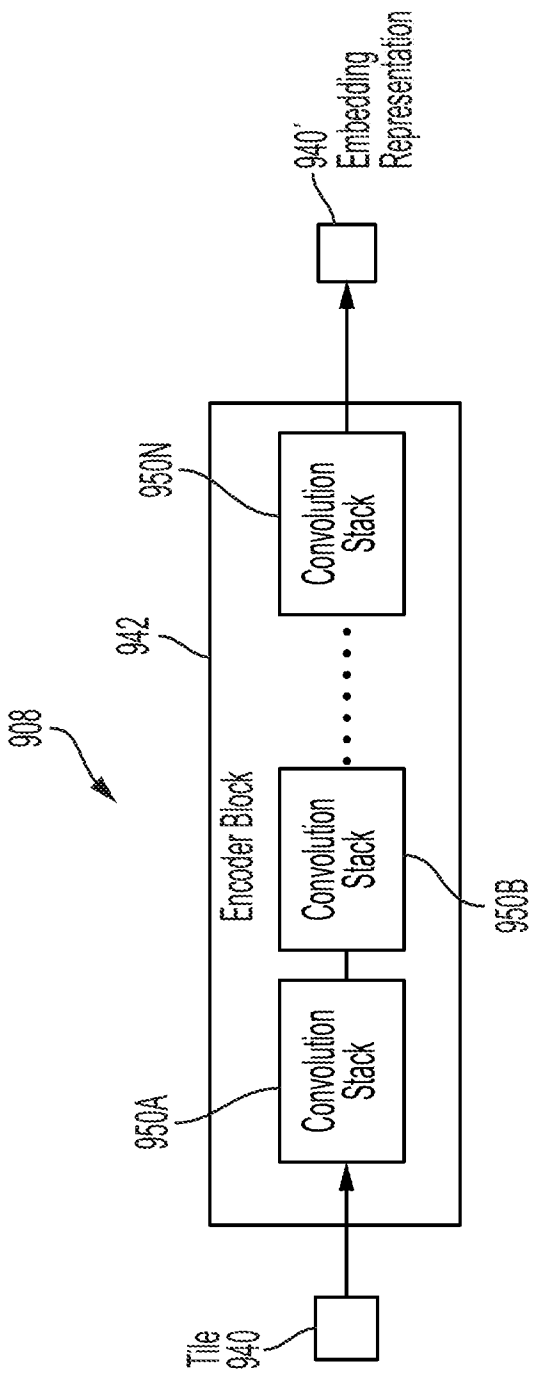
908
942
Encoder Block
950N
Convolution Stack
950B
Convolution Stack
950A
Convolution Stack
940'
Embedding Representation
Tile 940

FIG. 9C

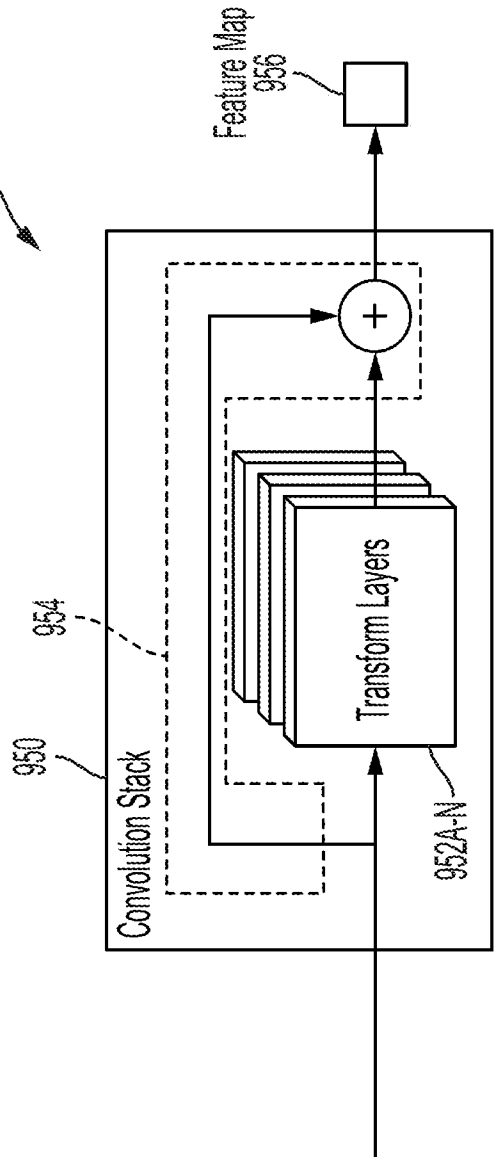
942
Feature Map 956
954
950
Convolution Stack
Transform Layers
952A-N

FIG. 9D

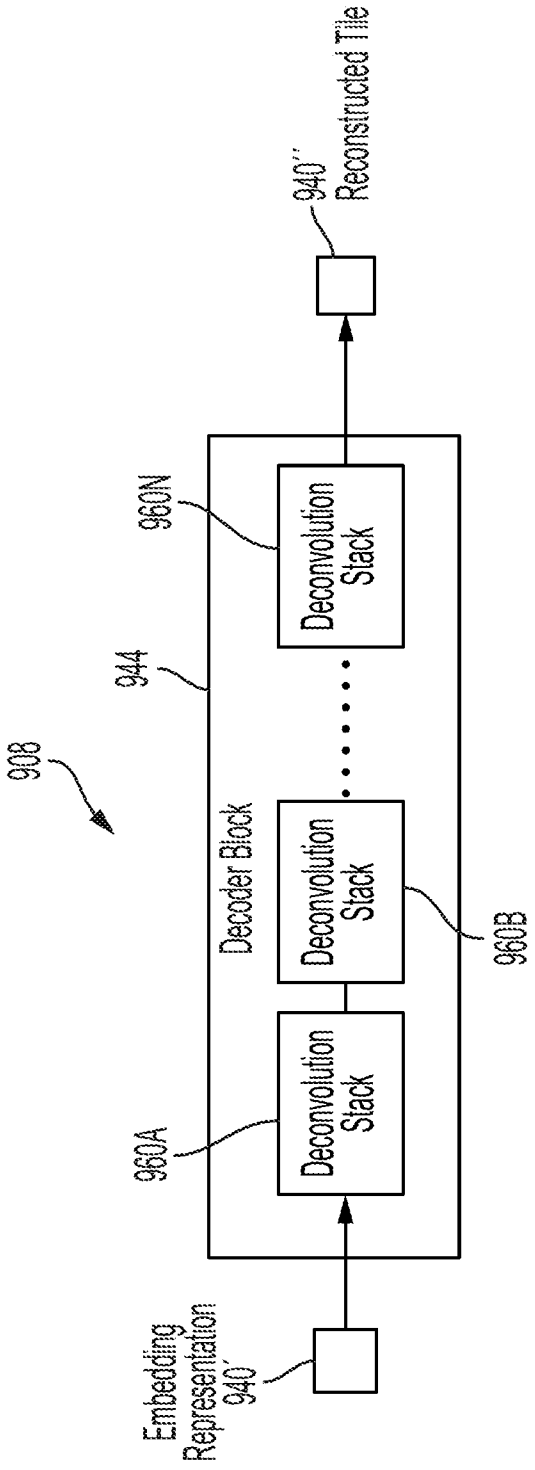
908
944
Decoder Block
960A
960B
960N
Deconvolution Stack
Deconvolution Stack
Deconvolution Stack
Embedding Representation 940'
940'' Reconstructed Tile

FIG. 9E

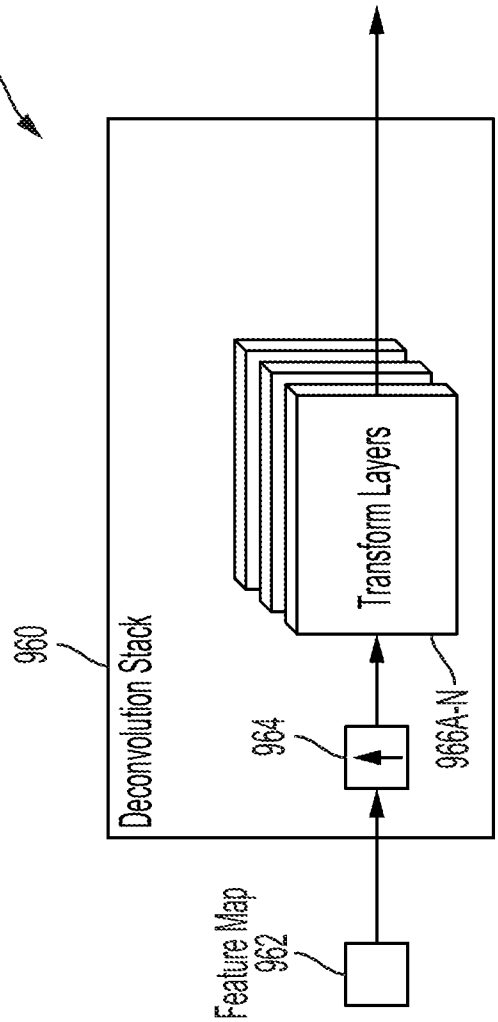
944
960
Deconvolution Stack
Transform Layers
964
966A-N
Feature Map 962

FIG. 9F

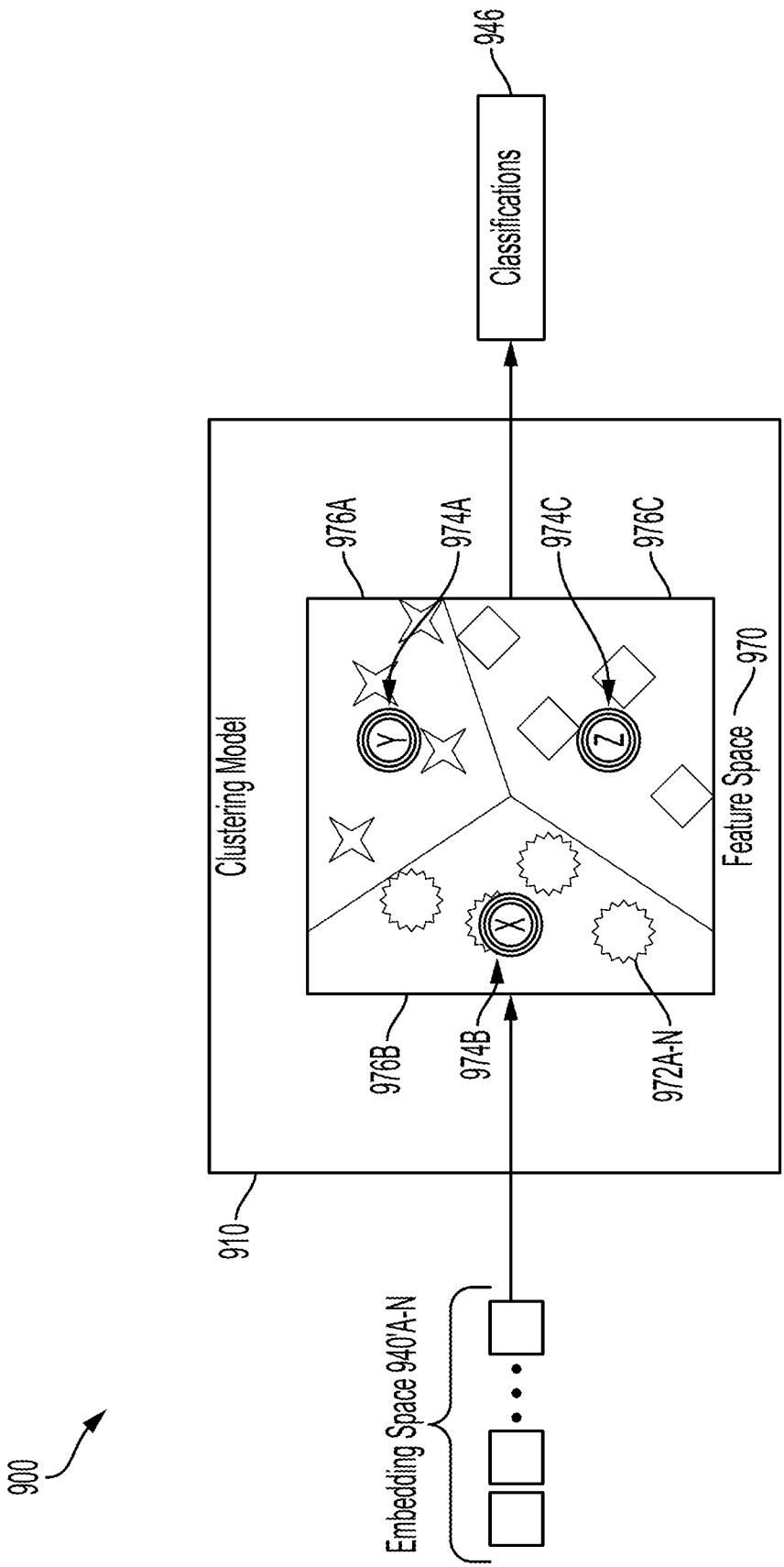
900
Embedding Space 940A-N
910
Clustering Model
976B
974B
972A-N
976A
974A
974C
976C
Feature Space
970
X
Y
Z
946
Classifications

FIG. 9G

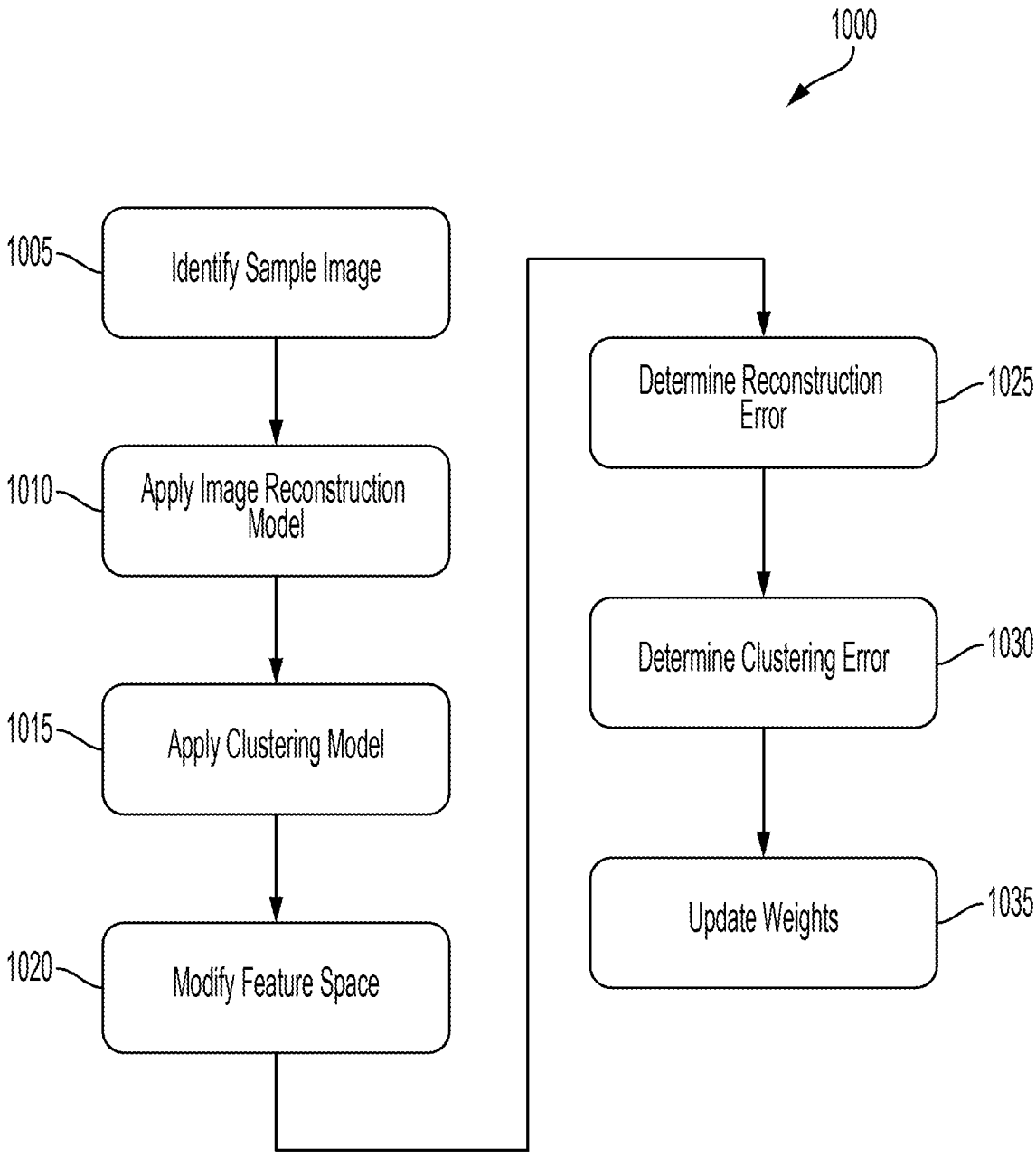
1000
1005
Identify Sample Image
1010
Apply Image Reconstruction Model
1015
Apply Clustering Model
1020
Modify Feature Space
1025
Determine Reconstruction Error
1030
Determine Clustering Error
1035
Update Weights

FIG. 10A

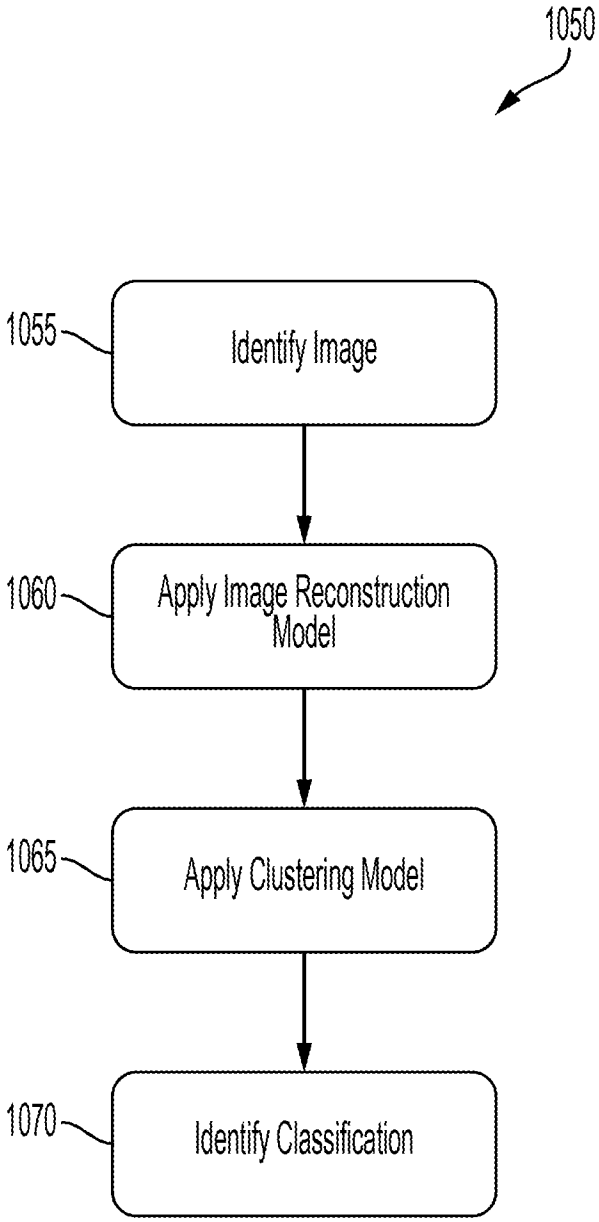
1050
1055
Identify Image
1060
Apply Image Reconstruction Model
1065
Apply Clustering Model
1070
Identify Classification

FIG. 10B

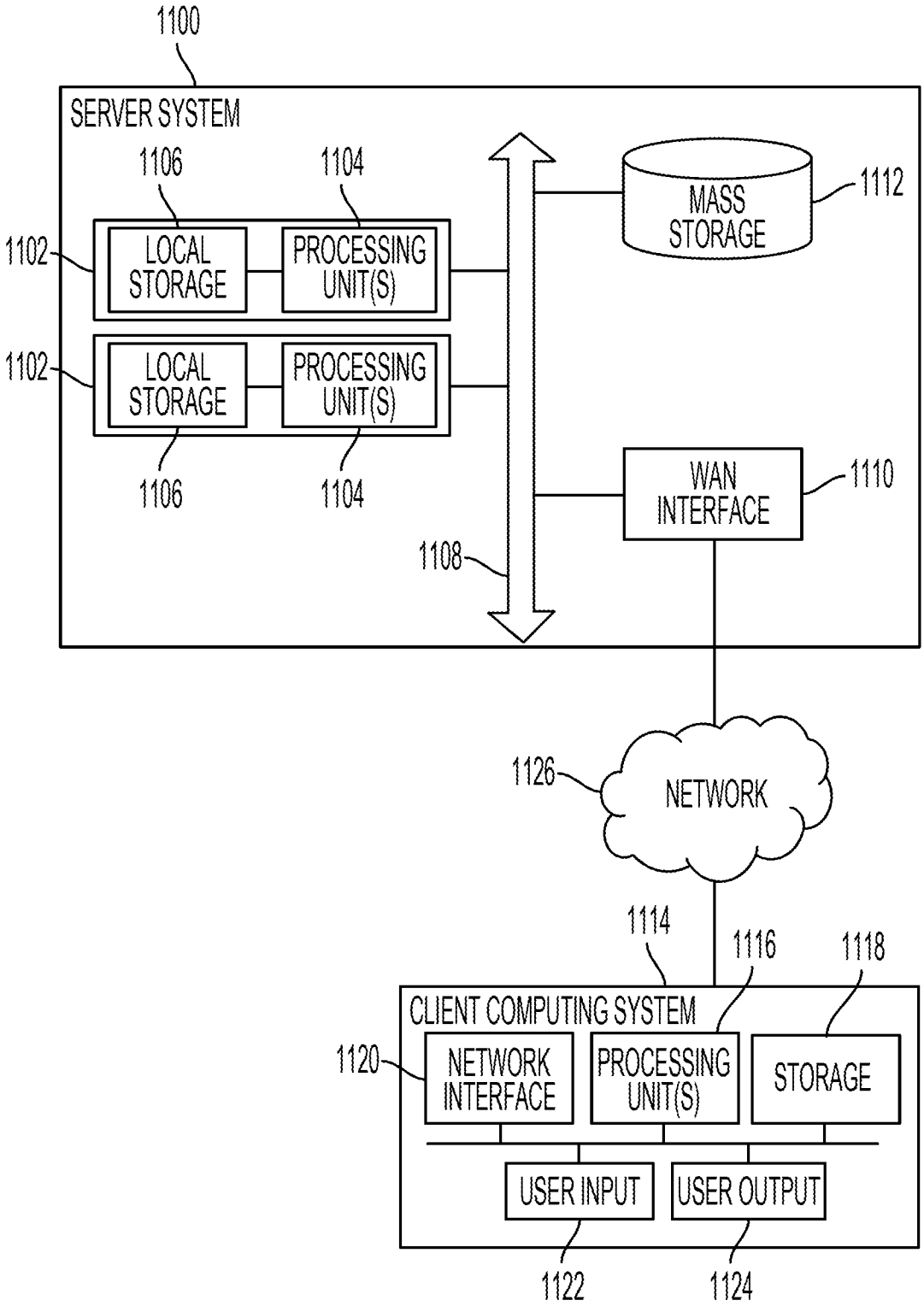
1100
SERVER SYSTEM
1106
1104
1102
LOCAL STORAGE
PROCESSING UNIT(S)
1102
LOCAL STORAGE
PROCESSING UNIT(S)
1106
1104
MASS STORAGE
1112
WAN INTERFACE
1110
1108
1126
NETWORK
1114
1116
1118
CLIENT COMPUTING SYSTEM
1120
NETWORK INTERFACE
PROCESSING UNIT(S)
STORAGE
USER INPUT
USER OUTPUT
1122
1124

FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/21210

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G06T 7/00 (2020.01)

CPC - G06T 7/0012, G06K 9/0014, G06K 9/00147, G06K 9/4652, G06T 2207/20021, G06K 9/6247, G06K 2209/05, G06T 2207/20016, G06T 2207/20081, G06T 9/002

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2016/0275374 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 22 September 2016 (22.09.2016) entire document (especially para [0046], [0064], [0074]-[0076], [0081], [0095], [0098], [0101])	1-20
A	US 2018/0374210 A1 (The Board Trustees of the Leland Stanford Junior University) 27 December 2018 (27.12.2018) entire document (especially para [0053]-[0055], [0089], [0091])	1-20
A	US 2016/0063724 A1 (PATHXL LIMITED) 03 March 2016 (03.03.2016) entire document (especially para [0031], [0033], [0039], [0051], [0114]	1-20
A	Lange et al. "Deep Auto-Encoder Neural Networks in Reinforcement Learning" [published online] The 2010 International Joint Conference on Neural Networks (IJCNN). IEEE, 2010. [Retrieved on 13 May 2020] Retrieved from < URL: http://ml.informatik.uni-freiburg.de/former/_media/publications/langeijcnn2010.pdf > pg. 1, col. 2; pg. 2, col. 2; pg. 5, col. 1; pg. 6, col. 1; Fig. 7	1-20
A	US 2018/0018451 A1 (Magic Leap, Inc.) 18 January 2018 (18.01.2018) entire document (especially para [0028]-[0029], [0041], [0043])	1-20
A	US 2016/0005183 A1 (Thiagarajan et al.) 07 January 2016 (07.01.2016) entire document	1-20

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

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 May 2020 (13.05.2020)	09 JUN 2020
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Lee Young Telephone No. PCT Helpdesk: 571-272-4300

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 20/21210

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018/0240233 A1 (Siemens Healthcare GmbH) 23 August 2018 (23.08.2018) entire document	1-20
A	WO 2018/211143 A1 (DEEPMIND TECHNOLOGIES LIMITED) 22 November 2018 (22.11.2018) entire document	1-20
A	Minnen, et al. "Spatially adaptive image compression using a tiled deep network." [published online] 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017. [Retrieved on 13 May 2020] Retrieved from < URL: https://arxiv.org/pdf/1802.02629.pdf > entire document	1-20
A	Wen, et al. "Deep residual network predicts cortical representation and organization of visual features for rapid categorization." [published online] Scientific reports 8.1 (2018): 1-17. [Retrieved on 12 May 2020] Retrieved from < URL: https://www.nature.com/articles/s41598-018-22160-9?dlv-emuid=&dlv-mlid=0 > entire document	1-20
A	US 2016/0358024 A1 (HYPERVERGE INC.) 08 December 2016 (08.12.2016) entire document	1-20
A	US 9,990,687 B1 (Deep Learning Analytics, LLC) 05 June 2018 (05.06.2018) entire document	1-20
A	US 2018/0267996 A1 (ADOBE SYSTEMS INCORPORATED) 20 September 2018 (20.09.2018) entire document	1-20