US 20250267283A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2025/0267283 A1**

QU et al. (43) **Pub. Date: Aug. 21, 2025**

(54) **METHOD FOR VIDEO-BASED PATCH-WISE VECTOR QUANTIZED AUTO-ENCODER CODEBOOK LEARNING FOR VIDEO ANOMALY DETECTION**

(71) Applicant: **Samsung Display Co., Ltd.**, Yongin-si (KR)

(72) Inventors: **Shuhui QU**, Fremont, CA (US); **Qisen CHENG**, Cupertino, CA (US); **Yannick BLIESENER**, San Jose, CA (US); **Janghwan LEE**, Pleasanton, CA (US)

(21) Appl. No.: **19/204,381**

(22) Filed: **May 9, 2025**

Related U.S. Application Data

(63) Continuation of application No. 18/074,195, filed on Dec. 2, 2022, now Pat. No. 12,301,833.

(60) Provisional application No. 63/395,782, filed on Aug. 5, 2022.

Publication Classification

(51) **Int. Cl.**

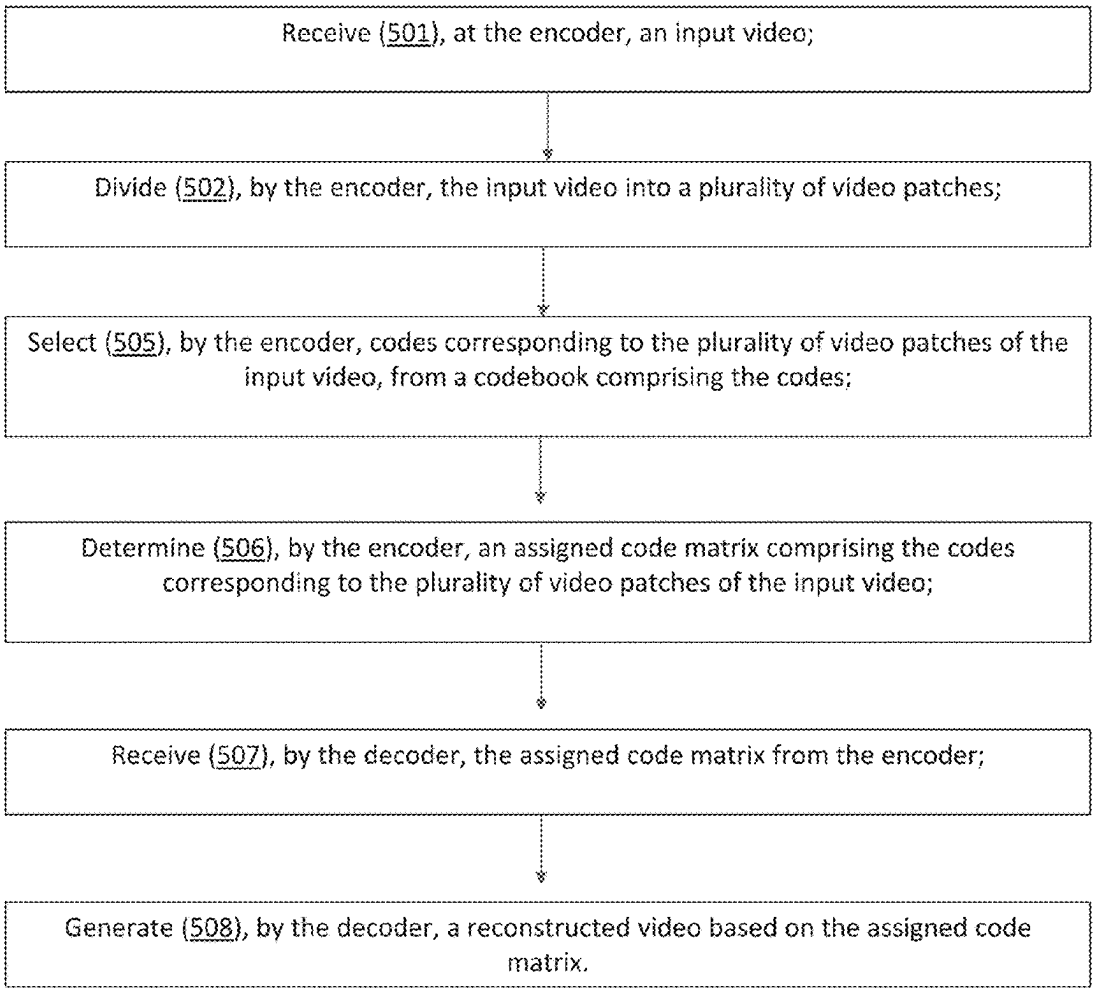
H04N 19/157	(2014.01)
G06V 10/44	(2022.01)
G06V 10/74	(2022.01)
G06V 20/40	(2022.01)
H04N 19/94	(2014.01)

(52) **U.S. Cl.**

CPC ***H04N 19/157*** (2014.11); ***G06V 10/44*** (2022.01); ***G06V 10/761*** (2022.01); ***G06V 20/49*** (2022.01); ***H04N 19/94*** (2014.11)

(57) **ABSTRACT**

According to some embodiments, a system includes: a memory, an encoder; a decoder, wherein the system is operable to: receive, at the encoder, an input video; divide, by the encoder, the input video into a plurality of video patches; select, by the encoder, codes corresponding to the plurality of video patches of the input video, from a codebook comprising the codes; determine, by the encoder, an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video; receive, by the decoder, the assigned code matrix from the encoder; and generate, by the decoder, a reconstructed video based on the assigned code matrix.



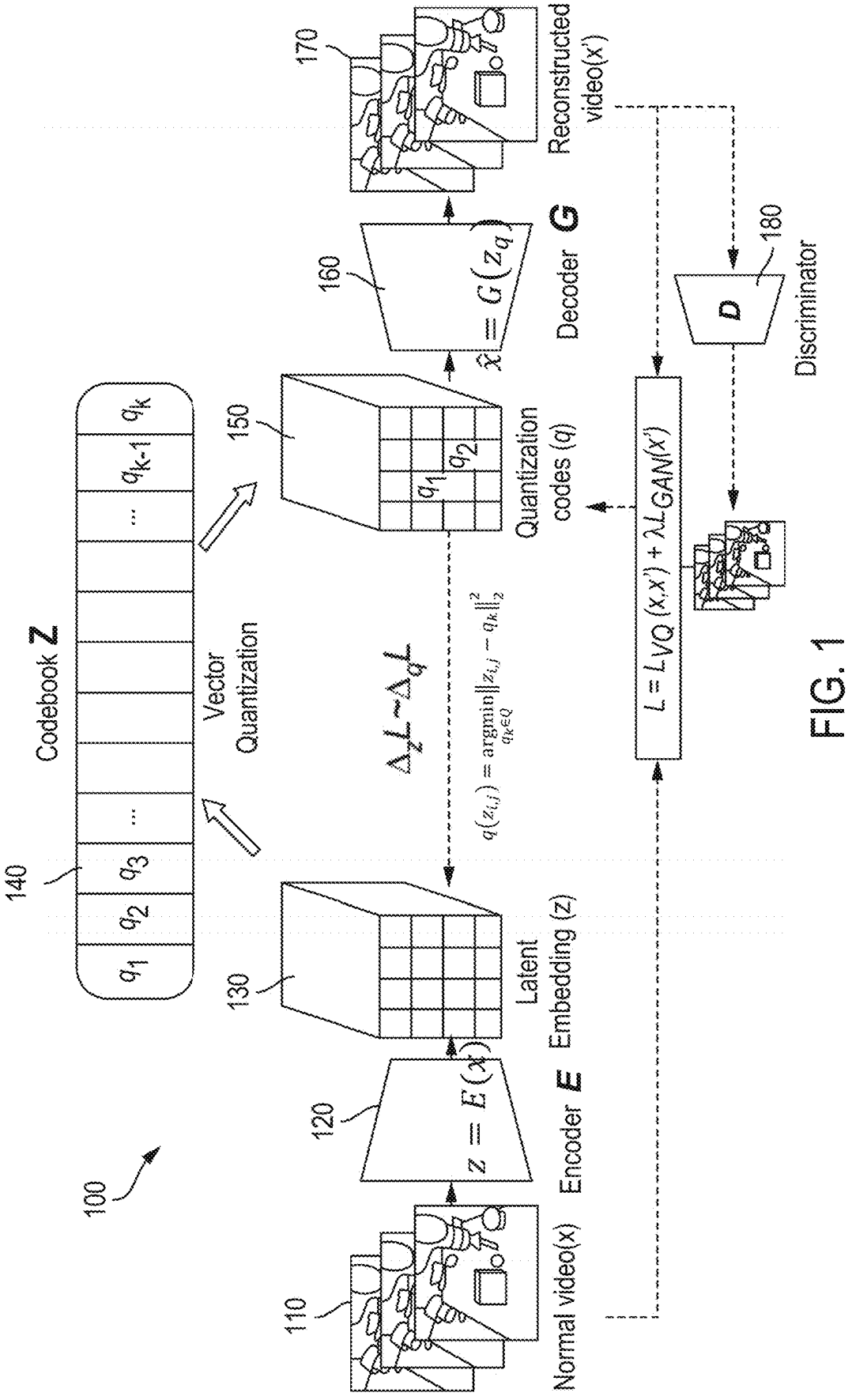

100
110
Normal video(x)
120
Encoder E
$z = E(x)$
130
Latent Embedding (z)
140
Codebook Z
q_1 q_2 q_3 ... q_{k-1} q_k
Vector Quantization
150
Quantization codes (q)
q_1 q_2
$q(z_{i,j}) = \underset{q_k \in Q}{\operatorname{argmin}} \|z_{i,j} - q_k\|_2^2$
$\Delta_z L \sim \Delta_q L$
160
Decoder G
$\hat{x} = G(z_q)$
170
Reconstructed video(x')
180
Discriminator
D
$L = L_{VQ}(x,x') + \lambda L_{GAN}(x')$

FIG. 1

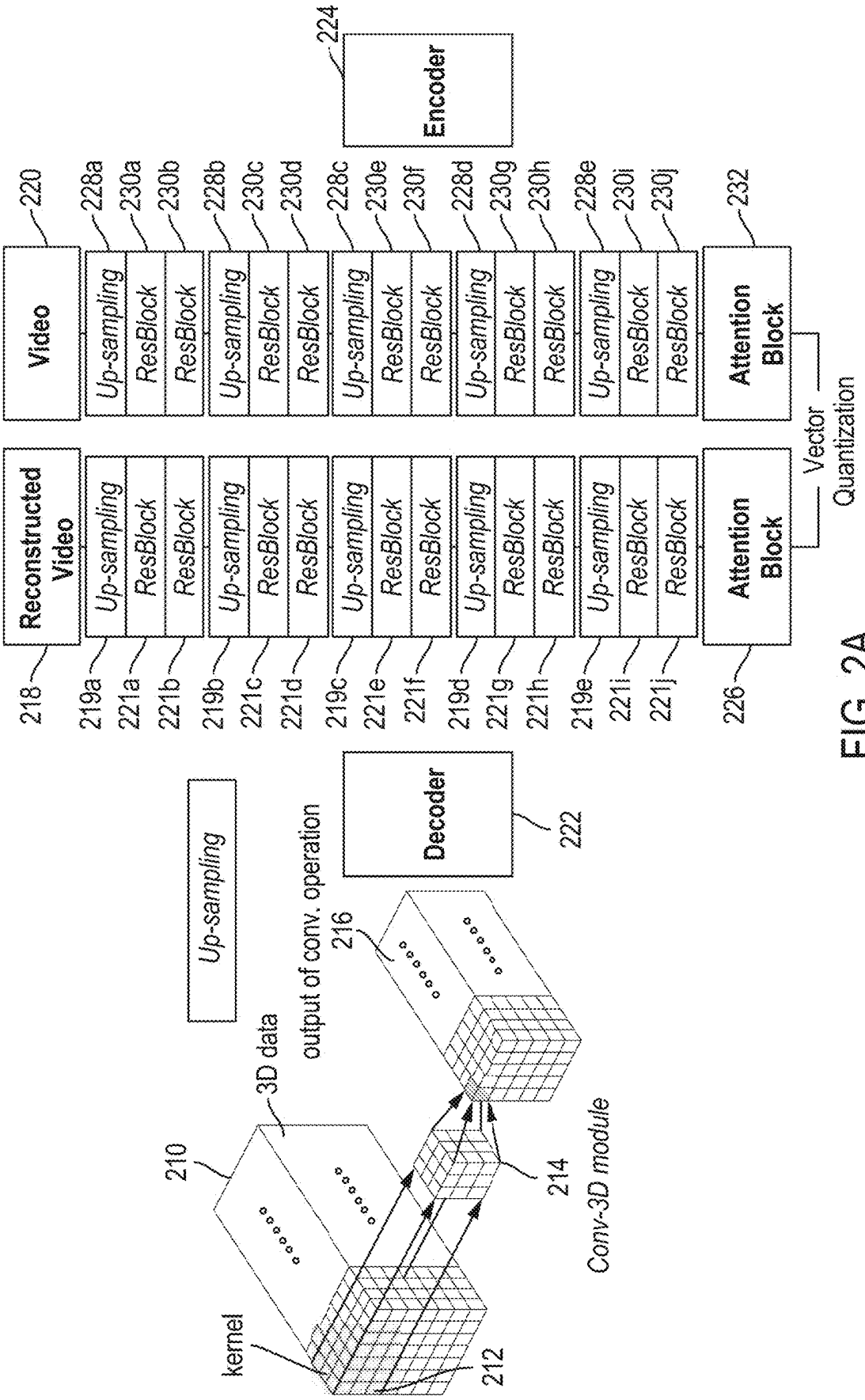

Encoder
224
Video
220
Reconstructed Video
218
Up-sampling
ResBlock
ResBlock
228a
230a
230b
228b
230c
230d
228c
230e
230f
228d
230g
230h
228e
230i
230j
Attention Block
232
Attention Block
226
Vector Quantization
219a
221a
221b
219b
221c
221d
219c
221e
221f
219d
221g
221h
219e
221i
221j
Decoder
222
Up-sampling
3D data
output of conv. operation
216
210
214
kernel
212
Conv-3D module

FIG. 2A

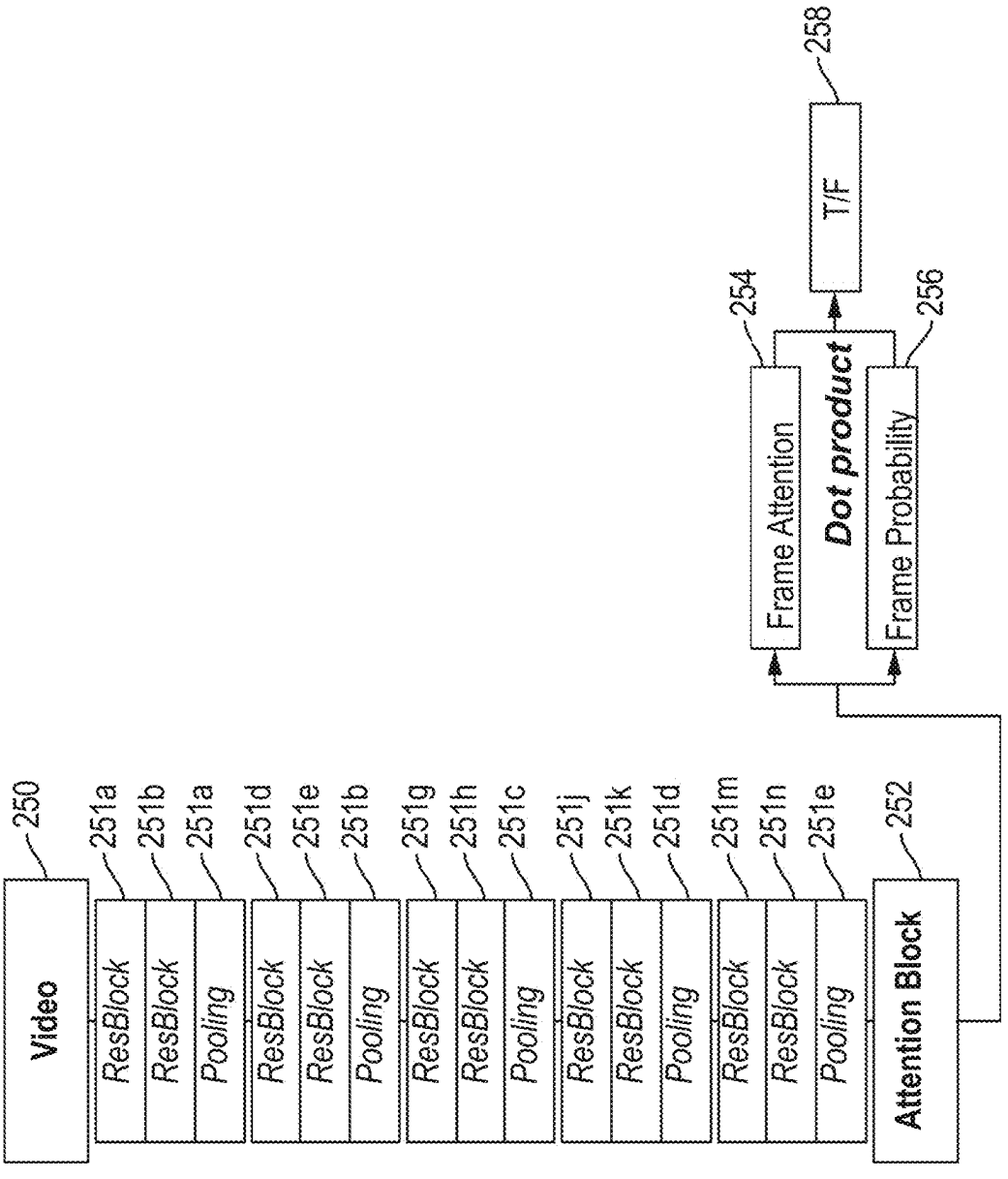

Video
250
ResBlock
ResBlock
Pooling
ResBlock
ResBlock
Pooling
ResBlock
ResBlock
Pooling
ResBlock
ResBlock
Pooling
ResBlock
ResBlock
Pooling
251a
251b
251a
251d
251e
251b
251g
251h
251c
251j
251k
251d
251m
251n
251e
Attention Block
252
Frame Attention
254
Dot product
Frame Probability
256
T/F
258

FIG. 2B

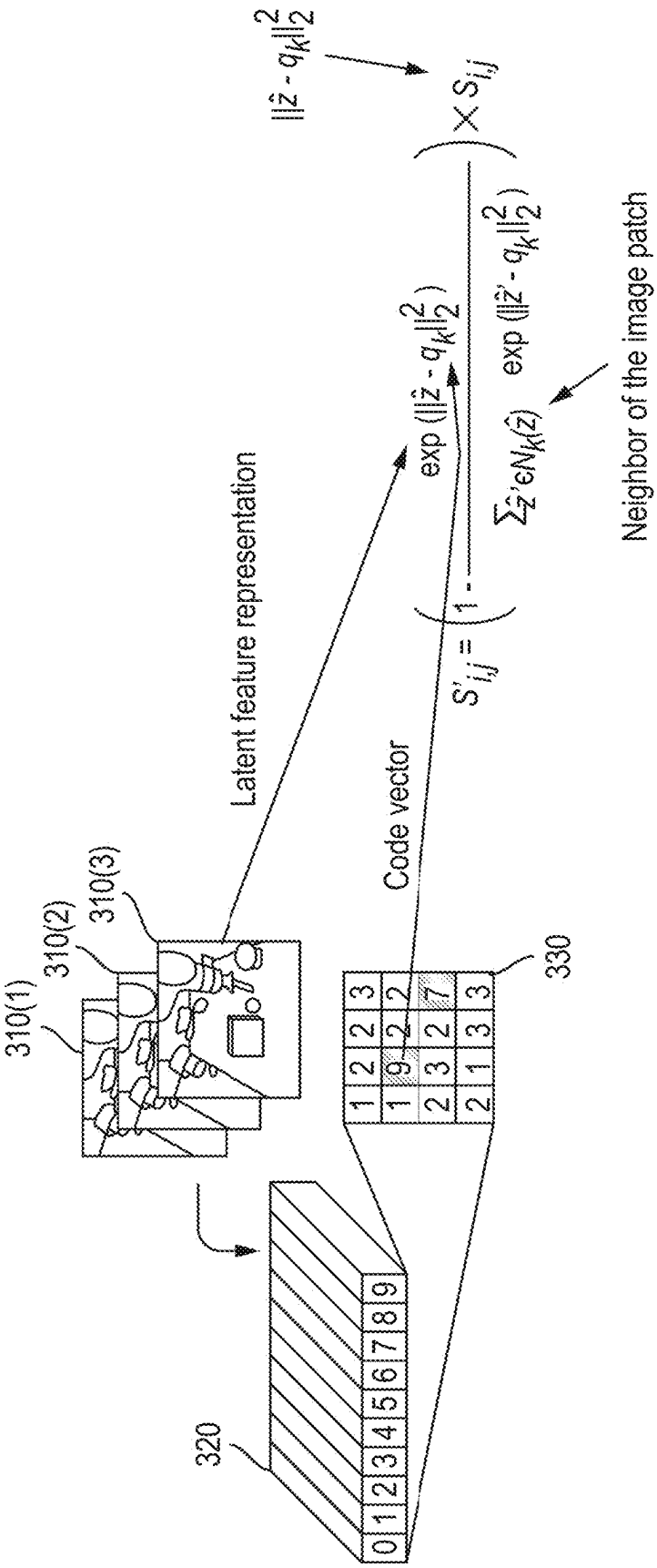

310(1)
310(2)
310(3)
Latent feature representation
320
0 1 2 3 4 5 6 7 8 9
Code vector
330
1 2 2 3
1 9 2 2
2 3 2 7
2 1 3 3
$S'_{i,j} = \left(1 - \frac{\exp(\|\hat{z} - q_k\|_2^2)}{\sum_{\hat{z}' \in N_k(\hat{z})} \exp(\|\hat{z}' - q_k\|_2^2)}\right) \times S_{i,j}$
$\|\hat{z} - q_k\|_2^2$
Neighbor of the image patch

FIG. 3

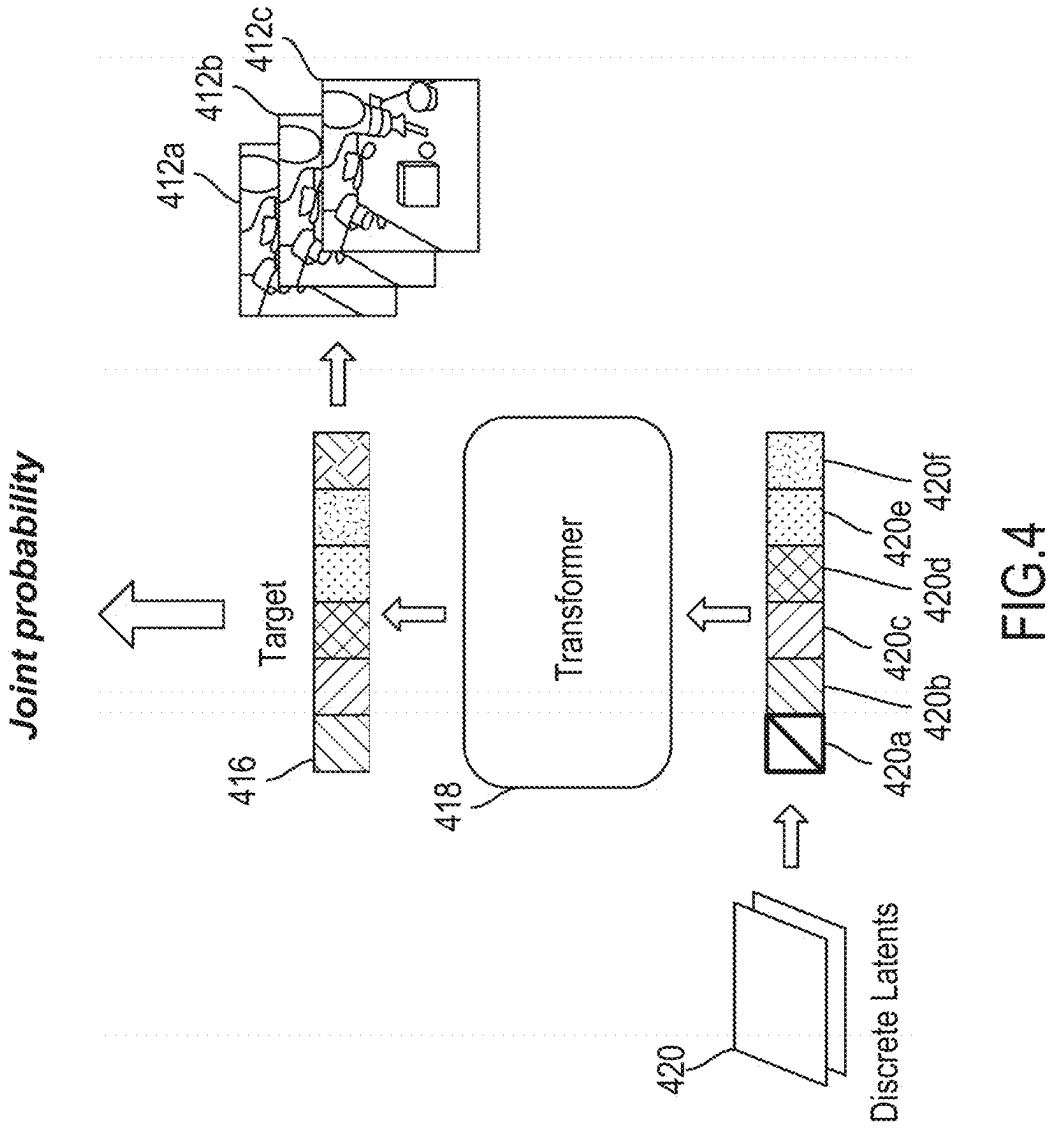

Joint probability
412a
412b
412c
Target
416
Transformer
418
420a
420b
420c
420d
420e
420f
420
Discrete Latents

FIG.4

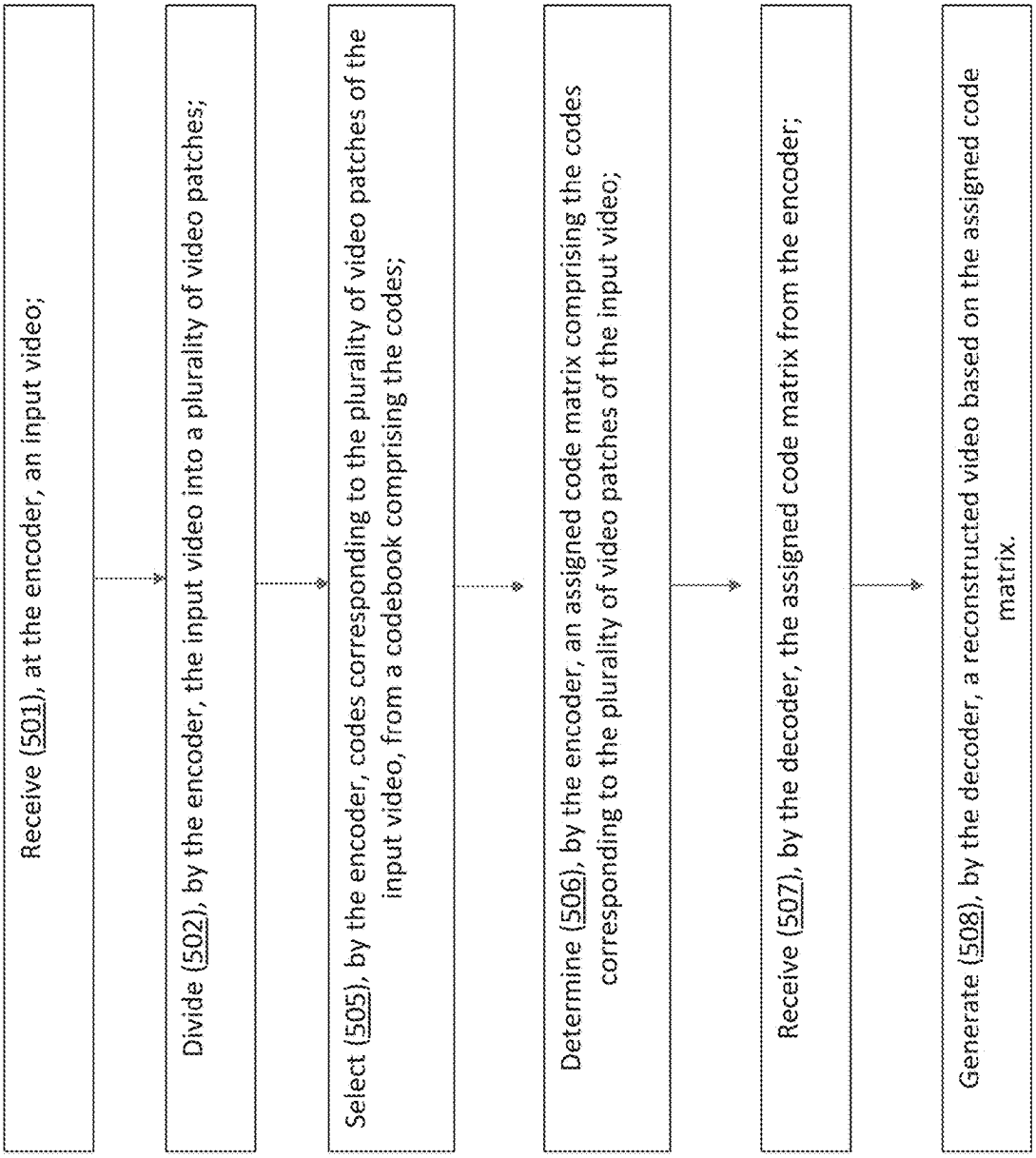

Receive (501), at the encoder, an input video;
Divide (502), by the encoder, the input video into a plurality of video patches;
Select (505), by the encoder, codes corresponding to the plurality of video patches of the input video, from a codebook comprising the codes;
Determine (506), by the encoder, an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video;
Receive (507), by the decoder, the assigned code matrix from the encoder;
Generate (508), by the decoder, a reconstructed video based on the assigned code matrix.

FIG. 5

METHOD FOR VIDEO-BASED PATCH-WISE VECTOR QUANTIZED AUTO-ENCODER CODEBOOK LEARNING FOR VIDEO ANOMALY DETECTION

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This is a continuation application of U.S. patent application Ser. No. 18/074,195, filed Dec. 2, 2022, which claims priority to and benefit of U.S. Provisional Patent Application No. 63/395,782 filed on Aug. 5, 2022, the entire contents of which are incorporated by reference herein.

FIELD

[0002] The present application generally relates to detecting anomalies, and more particularly to video-based vector quantized auto-encoder codebook learning for video anomaly detection.

BACKGROUND

[0003] In recent years, manufacturing factories are commonly monitored with hundreds or thousands of Closed Circuit Televisions (CCTVs) for production and infrastructure safety. Human-based CCTV anomaly detection is extremely tedious and time-consuming due to the pervasive use of surveillance cameras coupled with the growth of video data. Intelligent CCTV's powered with AI technology have been introduced in an attempt to reduce manual surveillance through automatic anomaly detection.

[0004] A first challenge with intelligent CCTV's powered with AI, among other aspects, are the extremely imbalanced data samples in a dataset with few, if any, anomaly videos. In this way, the ratio between "normal" samples and an "anomaly" samples are extremely imbalanced. Deep learning may require a balanced dataset to achieve an effective output or performance. Training on an imbalanced data would likely degrade the performance of deep learning and the predicted result would bias toward the normal sample.

[0005] A second challenge that intelligent CCTV's powered with AI presents is the low-resolution of anomalies within the CCTV frames. Tiny anomalies within the whole CCTV frame stream remains a detection issue.

[0006] As such, there exists a need to achieve a higher accuracy of CCTV anomaly detection using only normal video data and to achieve a higher accuracy of anomaly detection under varying resolution of anomalies.

[0007] The above information in the Background section is only for enhancement of understanding of the background of the technology and therefore it should not be construed as admission of existence or relevancy of the prior art.

SUMMARY

[0008] This summary is provided to introduce a selection of features and concepts of embodiments of the present disclosure that are further described below in the detailed description. This summary is not intended to identify key or essential features of the claimed subject matter, nor is it intended to be used in limiting the scope of the claimed subject matter. One or more of the described features may be combined with one or more other described features to provide a workable device.

[0009] Aspects of example embodiments of the present disclosure relate to vector quantized auto-encoder codebook learning for manufacturing display extreme minor defects detection.

[0010] (A1) In one or more embodiments, a system includes: a memory, an encoder, a decoder and a processor. The system is operable to receive, at the encoder, an input video, divide, by the encoder, the input video into a plurality of video patches, select, by the encoder, codes corresponding to the plurality of video patches of the input video, from a codebook comprising the codes, determine, by the encoder, an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video, receive, by the decoder, the assigned code matrix from the encoder; and generate, by the decoder, a reconstructed video based on the assigned code matrix.

[0011] (A2) The system of (A1), wherein the codes corresponding to the plurality of video patches of the input video are selected from the codebook using a look-up function, wherein the look-up function, the codebook, and the assigned code matrix are stored in the memory.

[0012] (A3) The system of (A2), wherein the system is further operable to: extract, by the encoder, latent features from the plurality of video patches of the input video; and determine, by the encoder, a latent features matrix comprising the latent features extracted from the plurality of video patches of the input video.

[0013] (A4) The system of (A3), in which the look-up function is operable to select a nearest code as an assignment for each of the latent features in the latent feature matrix to determine the assigned code matrix.

[0014] (A5) The system of (A3), wherein the encoder is operable to select a code corresponding to a video patch of the plurality of video patches of the input video from the codebook by comparing similarity measures between a latent feature representation of the video patch in the latent feature matrix and a corresponding code in the codebook.

[0015] (A6) The system of (A5), wherein the similarity measures comprise a Euclidean distance or a mahalanobis distance between the latent feature representation of the video patch in the latent feature matrix and the corresponding code in the codebook.

[0016] (A7) The system of (A1), wherein the assigned code matrix comprises the codes corresponding to the plurality of video patches of the input video.

[0017] (A8) The system of (A7), in which a code from among the codes in the assigned code matrix is assigned to a video patch of the plurality of video patches of the input video based on a vector quantization of a latent feature corresponding to the video patch.

[0018] (A9) The system of (A8), wherein a vector quantization loss of the system comprises a reconstruction loss that occurs during generation of the reconstructed video and a loss that occurs during the vector quantization of latent features corresponding to the plurality of video patches of the input video.

[0019] (A10) The system of (A9), further comprising: a patch-wise discriminator network operable to operate as a generative adversarial network to determine an adversarial training loss between the input video and the reconstructed video.

[0020] (A11) The system of (A10), wherein a total loss of the system in generating the reconstructed video from the input video comprises the vector quantization loss and the adversarial training loss.

[0021] (A12) The system of (A1), further operable to: receive, by the encoder, a test input video; divide, by the encoder, the test input video into a plurality of video patches; extract, by the encoder, latent features from the plurality of video patches of the test input video; encode, by the encoder, each of the plurality of video patches into a latent feature vector based on the extracted latent features; assign, by the encoder, a code to each of the plurality of video patches to determine assigned codes of the plurality of video patches, determine, by the encoder, a patch-set comprising the assigned codes,

[0022] determine, by the encoder, an anomaly score of each of the assigned codes of the patch-set, compare, by the encoder, the anomaly score of each of the assigned codes of the patch-set with a threshold and determine, by the encoder, a defect in one or more of the plurality of video patches based on a result of a comparison.

[0023] (A13) The system of (A12), wherein a code from the codebook that is of a shortest distance to the latent feature vector of a video patch of the plurality of video patches among the codes in the codebook is assigned to the video patch.

[0024] (A14) The system of (A12), wherein the anomaly score of each of the assigned codes of the patch-set is determined based on a probability density function.

[0025] (B1) A method comprising: receiving, at an encoder, an input video, dividing, at the encoder, the input video into a plurality of video patches, selecting, by the encoder, codes corresponding to the plurality of video patches of the input video, from a codebook comprising the codes; determining, by the encoder, an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video; receiving, by a decoder, the assigned code matrix from the encoder; generating, by the decoder, a reconstructed video based on the assigned code matrix.

[0026] (B2) The method of (B1), in which the codes corresponding to the plurality of video patches of the input video are selected from the codebook using a look-up function, wherein the method further comprises: extracting, by the encoder, latent features from the plurality of video patches of the input video; and determining, by the encoder, a latent features matrix comprising the latent features extracted from the plurality of video patches of the input video.

[0027] (B3) The method of (B2), wherein the look-up function is operable to select a nearest code as an assignment for each of the latent features in the latent feature matrix to determine the assigned code matrix, and wherein the encoder is operable to select a code corresponding to an video patch of the plurality of video patches of the input video from the codebook by comparing similarity measures between a latent feature representation of the video patch in the latent feature matrix and a corresponding code in the codebook.

[0028] (B4) The method of (B3), wherein a code from among the codes in the assigned code matrix is assigned to an video patch of the plurality of video patches of the input video based on a vector quantization of a latent feature corresponding to the video patch, wherein a vector quantization loss comprises a reconstruction loss that occurs during generation of the reconstructed video and a loss that occurs during the vector quantization of latent features corresponding to the plurality of video patches of the input video, and wherein a total loss for generating the reconstructed video from the input video comprises the vector quantization loss and an adversarial training loss.

[0029] (B5) The method of (B1), further comprising: extracting, by the encoder, latent features from the plurality of video patches of the input video, encoding, by the encoder, each of the plurality of video patches into a latent feature vector based on the extracted latent features, assigning, by the encoder, a code to each of the plurality of video patches to determine assigned codes of the plurality of video patches, determining, by the encoder, a patch-set comprising the assigned codes, determining, by the encoder, an anomaly score of each of the assigned codes of the patch-set, compare, by the encoder, the anomaly score of each of the assigned codes of the patch-set with a threshold and determine, by the encoder, a defect in one or more of the plurality of video patches based on a result of a comparison.

[0030] (C1) A non-transitory computer readable storage medium operable to store instructions that, when executed by a processor included in a computing device, cause the computing device to, receive, at an encoder of the computer device, an input video, divide, at the encoder, the input video into a plurality of video patches, select, by the encoder, codes corresponding to the plurality of video patches of the input video, from a codebook comprising the codes, determine, by the encoder, an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video, receive, by a decoder of the computer device, the assigned code matrix from the encoder; and generate, by the decoder, a reconstructed video based on the assigned code matrix.

[0031] (C2) A non-transitory computer readable storage medium operable to store instructions that, when executed by a processor included in a computing device, cause the computing device to perform any of (A2-14).

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] These and other features of some example embodiments of the present disclosure will be appreciated and understood with reference to the specification, claims, and appended drawings, wherein:

[0033] FIG. **1** illustrates a codebook learning of a video-based patch-wise vector-quantized auto-encoder (VBPVQAE) system, in accordance with some embodiments.

[0034] FIG. **2**A illustrates a block module for a 3D convolutional encoder backbone model of the VBPVQAE system of FIG. **1**, in accordance with some embodiments.

[0035] FIG. **2**B illustrates a block module for a 3D convolutional discriminator backbone model of the VBPVQAE system of FIG. **1**, in accordance with some embodiments.

[0036] FIG. **3** illustrates anomaly detection using patch-wise codebook learning, in accordance with some embodiments.

[0037] FIG. **4** illustrates anomaly detection using learned spatial-temporal dependency, in accordance with some embodiments.

[0038] FIG. **5** illustrates a method for anomaly detection using the VBPVQAE system, in accordance with some embodiments.

[0039] Aspects, features, and effects of embodiments of the present disclosure are best understood by referring to the detailed description that follows. Unless otherwise noted, like reference numerals denote like elements throughout the attached drawings and the written description, and thus, descriptions thereof will not be repeated. In the drawings, the relative sizes of elements, layers, and regions may be exaggerated for clarity.

DETAILED DESCRIPTION

[0040] The detailed description set forth below in connection with the appended drawings is intended as a description of some example embodiments of a system and a method for CCTV anomaly detection provided in accordance with the present disclosure and is not intended to represent the only forms in which the present disclosure may be constructed or utilized. The description sets forth the features of the present disclosure in connection with the illustrated embodiments. It is to be understood, however, that the same or equivalent functions and structures may be accomplished by different embodiments that are also intended to be encompassed within the scope of the disclosure. As denoted elsewhere herein, like element numbers are intended to indicate like elements or features.

[0041] It will be understood that, although the terms "first", "second", "third", etc., may be used herein to describe various elements, components, regions, layers and/or sections, these elements, components, regions, layers and/or sections should not be limited by these terms. These terms are only used to distinguish one element, component, region, layer or section from another element, component, region, layer or section. Thus, a first element, component, region, layer or section discussed herein could be termed a second element, component, region, layer or section, without departing from the scope of the present disclosure.

[0042] Spatially relative terms, such as "beneath", "below", "lower", "under", "above", "upper" and the like, may be used herein for ease of description to describe one element or feature's relationship to another element(s) or feature(s) as illustrated in the figures. It will be understood that such spatially relative terms are intended to encompass different orientations of the device in use or in operation, in addition to the orientation depicted in the figures. For example, if the device in the figures is turned over, elements described as "below" or "beneath" or "under" other elements or features would then be oriented "above" the other elements or features. Thus, the example terms "below" and "under" can encompass both an orientation of above and below. The device may be otherwise oriented (e.g., rotated 90 degrees or at other orientations) and the spatially relative descriptors used herein should be interpreted accordingly. In addition, it will also be understood that when a layer is referred to as being "between" two layers, it can be the only layer between the two layers, or one or more intervening layers may also be present.

[0043] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the present disclosure. As used herein, the terms "substantially," "about," and similar terms are used as terms of approximation and not as terms of degree, and are intended to account for the inherent deviations in measured or calculated values that would be recognized by those of ordinary skill in the art.

[0044] As used herein, the singular forms "a" and "an" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising", when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof. As used herein, the term "and/or" includes any and all combinations of one or more of the associated listed items. Expressions such as "at least one of," when preceding a list of elements, modify the entire list of elements and do not modify the individual elements of the list. Further, the use of "may" when describing embodiments of the present disclosure refers to "one or more embodiments of the present disclosure". Also, the term "exemplary" is intended to refer to an example or illustration. As used herein, the terms "use," "using," and "used" may be considered synonymous with the terms "utilize," "utilizing," and "utilized," respectively.

[0045] It will be understood that when an element or layer is referred to as being "on", "connected to", "coupled to", or "adjacent to" another element or layer, it may be directly on, connected to, coupled to, or adjacent to the other element or layer, or one or more intervening elements or layers may be present. In contrast, when an element or layer is referred to as being "directly on", "directly connected to", "directly coupled to", or "immediately adjacent to" another element or layer, there are no intervening elements or layers present.

[0046] Any numerical range recited herein is intended to include all sub-ranges of the same numerical precision subsumed within the recited range. For example, a range of "1.0 to 10.0" is intended to include all subranges between (and including) the recited minimum value of 1.0 and the recited maximum value of 10.0, that is, having a minimum value equal to or greater than 1.0 and a maximum value equal to or less than 10.0, such as, for example, 2.4 to 7.6. Any maximum numerical limitation recited herein is intended to include all lower numerical limitations subsumed therein and any minimum numerical limitation recited in this specification is intended to include all higher numerical limitations subsumed therein.

[0047] In some embodiments, one or more outputs of the different embodiments of the methods and systems of the present disclosure may be transmitted to an electronics device coupled to or having a display device for displaying the one or more outputs or information regarding the one or more outputs of the different embodiments of the methods and systems of the present disclosure.

[0048] The electronic or electric devices and/or any other relevant devices or components according to embodiments of the present disclosure described herein may be implemented utilizing any suitable hardware, firmware (e.g. an application-specific integrated circuit), software, or a combination of software, firmware, and hardware. For example, the various components of these devices may be formed on

one integrated circuit (IC) chip or on separate IC chips. Further, the various components of these devices may be implemented on a flexible printed circuit film, a tape carrier package (TCP), a printed circuit board (PCB), formed on one substrate or other appropriate architectures. Further, the various components of these devices may be a process or thread, running on one or more processors, in one or more computing devices, executing computer program instructions and interacting with other system components for performing the various functionalities described herein. The computer program instructions are stored in a memory which may be implemented in a computing device using a standard memory device, such as, for example, a random access memory (RAM). The computer program instructions may also be stored in other non-transitory computer readable media such as, for example, a CD-ROM, flash drive, or the like. Also, a person of skill in the art should recognize that the functionality of various computing devices may be combined or integrated into a single computing device, or the functionality of a particular computing device may be distributed across one or more other computing devices without departing from the spirit and scope of the exemplary embodiments of the present disclosure.

[0049] As discussed previously, there exists a need to achieve a higher accuracy of CCTV anomaly detection using only normal video data. Additionally, as discussed previously, there also exists a need to achieve a higher accuracy of anomaly detection rates under varying resolution of anomalies. This may be achieved, as described herein by (i) designing and building a patch-wise anomaly detection model to learn the pattern of normal scenarios, to improve the identification of tiny anomalies in local patches as (as shown in FIG. **1**) and the architecture of the learning models (as shown in FIGS. **2**A and FIG. **2**B), (ii) performing a detailed examination on each small patch and identifying whether anomalies exist or not as shown in FIGS. **3**, and (iii) performing temporal dependency analysis to find the correlation between patches across different time-frames as shown in FIG. **5**.

[0050] Table 1 depicts performance on existing datasets. "AUROC" is "area under the receiver operating characteristic." VBPVQAE has the highest performance, owing to the novel method presented herein. Specifically, VBPVQAE can identify the local anomaly together with anomalies that happens for a long duration (e.g., time dependent). However, the other methods including OCNN, OCELM and OCSVM cannot consider this anomaly pattern. OCELM refers to a One-class Extreme Learning Machine, OCNN refers to a One-class Neural Network and OCSVM refers to a One-class Support Vector Machine.

TABLE 1

AUROC	OCELM	OCNN	OCSVM	VBPVQAE
UCF-101	0.947	0.934	0.905	0.983

[0051] It should be noted that "anomaly" and "defect" may be used interchangeably herein. Object anomalies include, but are not limited to, fires, natural disasters, smoke scenarios and persistent anomaly objects such as waste, hazardous items, on the site scenarios. Activity anomalies include, but are not limited to, human misbehaviors. In some embodiments, this can be achieved by designing and building models to achieve high defect detection accuracy using only normal data samples. In other embodiments, this can be achieved by minimizing the impact of imbalanced distribution of normal and defect data on the deep learning model.

[0052] One or more embodiments of the present disclosure may provide a solution to the challenges in the defect detection using machine learning by a Video-based Patch-Wise Vector-Quantized Auto-Encoder (VBPVQAE) system that is trained on video. In one or more embodiments, the VBPVQAE system may mitigate the above mentioned challenges of defect detection. VBPVQAE divides the video frames into sequences of patches across time and learns a codebook from the video patches. A code in the codebook represents the feature pattern of a video patch. In this way, latent features from the video patches may be extracted. The codebook is learned through the vector quantization technique. Codes can then be assigned to each learned latent feature from codes assigned to the video patches, by the distance of the patch's latent features to the assigned code. VBPVQAE system can achieve state-of-the-art performance on defect detection.

[0053] Following the research of one-class methods, in one or more embodiments, the VBPVQAE system of the present disclosure may be trained without anomaly data. In one or more embodiments, the VBPVQAE system may use vector quantization. Vector quantization is a technique that enables the learning of expressive representations stored in a discrete codebook. In one or more embodiments, the context-rich codebook may contribute to state-of-the-art performance in many vision tasks, including video generation and synthesis. Moreover, the codebook can be easily integrated into generative models as an end-to-end framework. By introducing this technique to the one-class auto-encoder architecture, in one or more embodiments, the VBPVQAE system learns distinctive normal representations in an end-to-end fashion. In addition, the high expressivity of the learned codebook may allow the VBPVQAE system to be trained for multiple objects incorporating a vast span of visual patterns. As such, the VBPVQAE system of the present disclosure may eliminate the need for extra tuning and dataset-specific trade-offs and may alleviate the burden to train multiple models.

[0054] In one or more embodiments, with a convolutional neural network (CNN) as the encoder, each codebook entry may attend to localized visual patterns in a video. Unlike natural videos, videos captured in manufacturing settings usually involve complex yet similar patterns distributed across video constituents. Based on this observation, the retention of global information may be needed in codebook learning. In one or more embodiments, in case of vision transformer and variations, the multi-head attention (MHA) mechanism may have superior capability in capturing the relationship between long-range contexts. In one or more embodiments, MHA and CNN may be combined as the encoder backbone to enhance local and global information expression.

[0055] Based on the learned codebook, the one-class detection framework or VBPVQAE system of the present disclosure may calculate the anomaly score for each video patch by generating the posterior probability on retrieved codes, and identify the defect through the value of the anomaly score.

[0056] In one or more embodiments, the VBPVQAE system of the present disclosure consists of two parts: (1) a patch vector-quantization auto-encoder that learns a code-

book incorporating representations of the normal data, and (2) a defect detector that identifies defects based on the learned codebook. The VBPVQAE system of the present disclosure quantizes latent features into indices of a codebook. For example, the VBPVQAE system of the present disclosure may train an encoder network to build a discrete embedding table as well as a decoder network that utilizes the index in the embedding table for the reconstruction of each video patch (e.g., see FIG. **1**). Both the encoder and decoder may be convolutional neural networks (CNNs) and multi-head attention blocks (e.g., transformers). At test time, the encoder-decoder pair may utilize the learned codebook to generate an index matrix for all patches in the test video. By looking at the joint probability of codes corresponding to the indices, the defect detector calculates a score matrix that can be used for patch-wise defect identification.

[0057] As such, one or more embodiments of the present disclosure provide a VBPVQAE system for one-class defect detection problems in manufacturing. The VBPVQAE system of the present disclosure may be trained on normal data and critically, under varying resolution of anomalies, hence largely reducing the need for defect samples at the training time. The principle of the VBPVQAE system is based on learning and matching normal representations in a learned codebook. By leveraging the vector quantization technique, codebook learning in an end-to-end fashion may be achieved. VBPVQAE system of the present disclosure may eliminate complicated tuning or trade-offs that are specific to different datasets. In one or more embodiments, representation learning of global context may also be enhanced by combining the CNN encoder and multi-head self-attention mechanism. This approach may lead to a successful encoding of a wide range of visual patterns. In one or more embodiments, the high expressivity of the codebook may help to achieve representation learning for different objects using a single model. In addition, in one or more embodiments, the context-rich codebook may further benefit other downstream tasks such as video reconstruction or synthesis of normal videos.

[0058] FIG. **1** illustrates a codebook learning of the VBPVQAE system of the present disclosure.

[0059] VBPVQAE system **100** quantizes latent features into a codebook of entries, (i.e. a latent feature matrix) with a pair of encoder (E) and decoder (G) networks. For example, VBPVQAE system **100** uses vector quantization to learn a codebook for the whole dataset, thus the dataset can be expressed by the codebook. In one or more embodiments, a code represents the display panel's video patches' features.

[0060] For example, at training, the encoder (E) **120** reads in videos (e.g., video (x) **110**) and extracts and stores visual patterns into a codebook (Q) **140**, whereas the decoder (G) **160** aims to select a proper code index for each video patch and tries to reconstruct the input videos based on the selected codes.

[0061] In one or more embodiments, the latent embedding codebook (Q) **140** may be represented as $Q \in R(K \times nz)$, where K is the number of the discrete codes in the codebook, and nz is the dimensionality of the latent embedding vector $qk \in R(nz)$, $k \in 1, 2, \ldots, K$.

[0062] For example, during training of the VBPVQAE system **100**, as shown in FIG. **1**, the encoder (E) **120** receives a video (x) **110**, where $x \in R(H \times W \times Tx3)$ and outputs the latent feature matrix (z) **130**. Here, H is the height of the video (x) **110**, W is the width of the video (x) **110**, T is the time of video (x) **110** and number "3" represents the number of channel of the video (x) **110** (e.g., red, green, blue).

[0063] For example, in the method of FIG. **1**, when the input video (x) is received at the encoder (E), the input video (x) may be divided into a plurality of video patches. In one or more embodiments, a latent feature from each of the plurality of patches may be extracted to determine the latent feature matrix (z) **130**, where $z \in R(h \times w \times nz)$ and $z = E(x)$. Here, h is the height of the latent feature matrix (z) **130**, w is the width of the latent feature matrix (z) **130**, and nz represents the hidden vector size of the latent feature matrix (z) **130**.

[0064] For example, after the latent feature matrix (z) **130** is determined by the encoder (E) **120**, vector quantization may be used to learn the codebook (Q) **140** to assign a code from the codebook (Q) **140** to each of the plurality of latent features of the latent feature matrix (z). Here, each of the latent features in the latent feature matrix (z) corresponds to each patch of the plurality of patches from the input video (x). In some embodiments, a patch is a sub-tensor of the video tensor. For example a video tensor matrix may be defined as $V \in R^{H,W,T,C}$, where H is height, W is width, T is time, C is channel. A patch $V_{h:h+\delta h,w:w+\delta w,t:t+\delta t,c}$ is part of the video tensor matrix, V, where δh, δw, δt refers to the corresponding step size.

[0065] In one or more embodiments, the VBPVQAE system **100** may further include a memory and a processor, and the latent feature matrix (z) **130** may be determined by the processor based on the latent features extracted from the plurality of video patches and in one or more embodiments, the latent feature matrix (z) **130** may be stored in the memory of the VBPVQAE system **100**.

[0066] For example, after the latent feature matrix (z) **130** is determined by the encoder (E) **120**, in the forward pass, a code for each video patch is selected (or assigned) from the codebook (Q) **140** using a look-up function q(.) by the encoder (E) **120** or the processor of the VBPVQAE system **100**. In one or more embodiments, the look-up function q(.) and the codebook (Q) **140** may be stored in the memory of the VBPVQAE system. By comparing the similarity measures (e.g., Euclidean distance, mahalanobis distance, etc.) between the latent feature representation of an input video patch in the latent feature matrix (z) **130** and each code in the codebook (Q) **140**, the look-up function q(.) selects the nearest code as an assignment for each latent feature in the latent feature matrix (z) **130** to generate the assigned code matrix (q) **150**. For example, in one or more embodiments, by comparing the Euclidean distance between the latent feature representation of an input patch in the latent feature matrix (z) **130** and each code in the codebook (Q) **140**, the look-up function q(.) selects the nearest code as an assignment for each latent feature in the latent feature matrix (z) **130**. In one or more embodiments, the assigned code matrix (q) **150** may be generated by the encoder (E) **120** or by the processor of the VBPVQAE system **100**.

[0067] For example, an assignment in the assigned code matrix (q) **150** may be represented as:

$$z_q^{i,j} = q(z_{i,j}) = \operatorname{argmin} \| z_{i,j} - q_k \|_2^2 \tag{1}$$

[0068] For example, $z_q^{i,j}$ may represent an element of the assigned code matrix (q) **150**.

[0069] For example, in one or more embodiments, a code may be assigned to each input video patch based on vector quantization of the latent feature corresponding to the input video patch. For example, in the codebook learning of the VBPVQAE system **100**, the concept of vector quantization may be used while determining the assigned code matrix (q) **150** by selecting a code for each video patch from the codebook (Q) **140** with a look-up function q(.).

[0070] The assigned code matrix (q) **150** is then passed to the decoder (G) **160** to generate a reconstructed video or a reconstructed input video ($\hat{x}$) from the elements of the assigned code matrix (q) **150**. In one or more embodiments, the assigned code matrix (q) **150** has the same shape as the latent feature matrix (z) **130**. For example, the assigned code matrix (q) **150** may be the quantized version of the latent feature matrix (z) **130**. The assigned code matrix (q) **150** is then passed into the decoder (G) **160**, which is an unsampling network that takes in the hXwXnz feature matrix and output HXWX3 video.

[0071] In one or more embodiments, the reconstructed input video ($\hat{x}$) may be represented as:

$$\hat{x} = G(z_q) = G(q(E(x))) \tag{2}$$

[0072] The difference between reconstructed video ($\hat{x}$) and the corresponding input video x is used as the guidance for the VBPVQAE system **100**.

[0073] In one or more embodiments, the argmin operand is not differentiable during backpropagation. The gradient of this step may be approximated by using the straight-through estimator (STE) and directly pass the gradients from q(z) to z, so that the reconstruction loss may be incorporated with the loss for the neural discrete representation learning. As such, the vector quantization loss function may be represented as:

$$L_{VQ}(E, G, Z) = \|\hat{x} - x\|_2^2 + \|sg[E(x)] - q\|_2^2 + \beta\|sg[E(x)] - q\|_2^2 \tag{3}$$

[0074] In equation (3), the first term $\|\hat{x}-x\|_2^2$ represents the reconstruction loss (e.g., the loss during the generation of the reconstructed video ($\hat{x}$)). Also, in equation (3), the second and third terms $\|sg[E(x)]-q\|_2^2+\beta\|sg[E(x)]-q\|_2^2$ represent vector quantization losses.

[0075] Further, in equation (3), the sg function sg[E(x)] represents the stop gradient operand that enables the zero partial derivatives, q is the code embedding, and β is the hyper-parameter.

[0076] In one or more embodiments, adversarial training may be incorporated in the codebook learning to enhance the expressivity of learned embedding's in the latent feature space. For example, in one or more embodiments, as shown in FIG. **1**, a patch-wise discriminator network (D) **180** may be added to the encoder-decoder framework (e.g., including the encoder (E) **120** and decoder (G) **160**) as in the generative adversarial network (GAN). An adversarial training loss L_{GAN} may also appended to encourage the VBPVQAE system **100** to differentiate between the original video (x) **110** and the reconstructed video ($\hat{x}$) **170**. In one or more embodiments, the patch-wise discriminator network (D) **180** may determine the adversarial training loss L_{GAN} and the total loss L of the VBPVQAE system **100**.

[0077] The adversarial training loss between the original input video (x) **110** and the reconstructed video ($\hat{x}$) **170** may be represented as:

$$L_{GAN}((E, G, Q), D) = \log(D(x)) + \log(1 - D(\hat{x})) \tag{4}$$

[0078] As a whole, the total loss of the VBPVQAE system **100** training is:

$$L = \text{argmin}_{E,G,Q} \max_D L_{VQ}(E, G, Z) + \lambda L_{GAN}((E, G, Q), D) \tag{5}$$

[0079] In one or more embodiments, equation (5) or the term "L" may represent the total loss of the VBPVQAE system **100** during training. For example, the term "L" in equation (5) essentially incorporates the vector quantization loss "L_{VQ}(E, G, Z)" and the GAN loss or the adversarial training loss between the original video (x) **110** and the reconstructed video ($\hat{x}$) **170** "L_{GAN}((E, G, Q), D)". In equation (5) "λ" is the hyperparameter that can be adaptively tuned.

[0080] In one or more embodiments, real-world videos from production line may incorporate sophisticated visual patterns that contain subtle defective traits. In order to learn an effective codebook from such complex datasets, it may be desirable to adopt a model that can concurrently (e.g., simultaneously) learn local features and understand the global composition of these local features. By encoding the information of both perspectives, the product video can be represented by a series of locally vivid as well as globally coherent perceptional codes. In order to achieve this, on top of convolutional layers, the multi-head self-attention layer may be adopted to learn the inter-correlation dependencies between elements within a sequence (i.e. words for language tasks or video patches for vision tasks). As a consequence, the VBPVQAE system **100** can learn a codebook with richer perceptions of complex industrial products.

[0081] FIG. **2**A is a block module representation for an example 3D convolutional encoder backbone model and FIG. **2**B is a block module for an example 3D convolutional discriminator backbone model of the VBPVQAE system of FIG. **1**, in accordance with some embodiments. It is to be noted, that model, as used herein, can also be interpreted as system architecture.

[0082] Input sensor **210** represents the input to the block module. Sub-tensor **212** is a sub-tensor of the input sensor **210** and represents the local representation of the input sensor **210**. Convolutional kernel **214** convolutes the sub-tensor **212**.

[0083] Output **216** is the output of convolute operation **212** and **214**. Decoder **222** is the backbone to reconstruct the original videos. Decoder **222** reconstructs the original videos from the quantized compressed representation.

[0084] Reconstructed video **218** is the output of the decoder **222**. Reconstructed video **218** reconstructs the original input video **220**, as explained in FIG. **1**. Up sampling blocks **219** *a-e* may include bilinear sampling, interpolation sampling, etc., to up sample from lower resolution, few frames, to higher resolution and more time frames.

[0085] Residual Block **221***a-j* are Residual Network 3D blocks, which constitute the residual connection of the block module. Residual Block **221** *a-j* function to convolute the input tensors. Attention block **226** is a multi-head attention block. It is used to redistribute the tensor weight along both spatial and temporal dimensions according to the learned importance of the tensor. Video **220** is input video, which, in some embodiments, provides the information/pattern to be learned by the neural networks. Down-sampling blocks **228** *a-e* of video **220** compress the dimension of the tensors (i.e. through the epooling layer, including maxpooling, average pooling and etc.).

[0086] Residual Block **230** *a-j* are Residual Network 3D blocks as well. Residual Block **230** *a-j* convolute the input tensors. Attention block **232** is a multi-head attention block to redistribute the weight along spatial and temporal dimensions.

[0087] FIG. **2**B is a block module for the 3D convolutional discriminator backbone model which helps identify if a video is from the reconstructed video **218** or the original video **220**, as shown in FIG. **2**A, resulting in an improvement of the encoder and decoder training.

[0088] Video **250** may be the original video or reconstructed vector quantized video.

[0089] Residual block **251** *a-n* are Residual Network 3D blocks as well. In some embodiments, residual blocks **251** *a-n* convolute the tensors. Pooling **251***a-e* are the down-sampling layers. In some embodiments, pooling **251** *a-e* down-sample the input tensor, which may include max-pooling, average pooling, etc.

[0090] Frame attention **254** is the weight assigned to each frame, (i.e. contribution of each frame for the final prediction). Frame probability **256** is the probability of each frame, whether the input tensor is reconstructed or original. Output **258** is the output of the network, which predicts whether the input video is reconstructed or original.

[0091] In one or more embodiments, VBPVQAE system **100** may also determine defects in a system. For example, FIG. **3** illustrates defect detection using video-based patch-wise codebook learning. For example, referring back to FIG. **1**, given an input video, (x), at the test phase, the trained VBPVQAE system **100** may identify defects by estimating the anomaly score $s_{i,j}$ for each video patch (e.g., **310**(**1**), **310**(**2**), **310**(**30**) . . . , **310**(*n*)) indexed at i, j from the assigned codes z_q.

[0092] As in the training phase, in the embodiment of FIG. **3**, an input video, (x), at the test phase may be divided into a plurality of video patches. Next, latent features from the plurality of video patches of the input video (x) at test may be extracted, and a video patch of the plurality of video patches may be encoded into the latent feature vector $\hat{z}$ (shorthand for $z_{i,j}$) by the encoder (E) **120**.

[0093] In FIG. **3**, the elements **310**(**1**), **310**(**2**), **310**(**3**) . . . represent the latent feature vector $\hat{z}$ representation of the video patches of the input video x at the test phase. Then k^{th} code from the codebook (Q) **320** that is of the shortest distance to the latent feature vector $\hat{z}$ among all the codes in the codebook (Q) **320** (e.g., based on the similarity measure between the latent feature vector $\hat{z}$ representation of the input video patch and the k^{th} code from the codebook (Q) **320**) is assigned to the video patch (e.g., **310**(**1**), **310**(**2**), **310**(**3**) . . . , **310**(*n*)) by the encoder (E) **120**. In one or more embodiments, the k^{th} code from the codebook (Q) **140** may be represented as:

$$k = \text{argmin}_{k \in 1,2,\ldots,K} \|\hat{z} - q_k\|_2^2 \tag{6}$$

[0094] With all video patches, the latent feature vector $\hat{z}$ is extracted and a patch-set **330** of the corresponding codes is determined by the encoder (E) **120**. The patch-set **330** of the corresponding codes may be represented as:

$$\{\hat{z} \mid \text{argmin}_{k \in 1,2,\ldots,K} \|\hat{z} - q_k\|_2^2 = k\}_{k=1,2,\ldots,K} \tag{7}$$

[0095] In one or more embodiments, the patch-set **330** may include the assigned codes corresponding to each of the plurality of video patches (e.g., **310**(**1**), **310**(**2**), **310**(**3**).

[0096] For each entry in the patch-set **330**, an anomaly score $s_{i,j} \in R$ may be calculated using either the probability density function $s_{i,j}=P(\hat{z})$, or weighted distance to the k nearest-neighbors in the patch-set **330**. For example, in FIG. **3**, an anomaly scoring matrix(s) **340** incorporates the anomaly score $s_{i,j}$ for each entry in the patch-set **330**. In one or more embodiments, the anomaly score $s_{i,j}$ may be represented as:

$$s_{i,j} = \left(1 - \frac{\exp(\|\hat{z} - q_k\|_2^2)}{\sum_{\hat{z}' \in N_k(\hat{z})} \exp(\|\hat{z} - q_k\|_2^2)}\right) \times s_{i,j} \tag{8}$$

[0097] In one or more embodiments, the patch-wise anomaly score $s_{i,j}$ may be reorganized into the anomaly scoring matrix(s) **340** according to the spatial location of the anomaly scores $s_{i,j}$. In one or more embodiments, the anomaly score $s_{i,j}$ may be calculated by the encoder (E) **120**.

[0098] In one or more embodiments, during the defect detection, an input sample is determined to be defective if any of the anomaly scores $s_{i,j}$ is larger than a given threshold at index k, which may be represented as:

$$\Pi(s_{i,j} > t_k) \forall\, s_{i,j} \in s \tag{9}$$

[0099] FIG. **4** illustrates defect detection with learned spatial-temporal dependency. Discrete latents **420** *a-f* represent all frames local patches' feature representation along spatial and temporal dimensions through the learned code. Transformer **418** learns the spatial and temporal correlation between these codes from normal videos, and takes in the sequence of latent codes to learn the correlation through self-supervised learning tasks including predicting the masked code, predicting next sequence and etc. The output target's probability presents how normal the input sequence are. Output sequence **416***a-f* are the output sequence from the transformer. Output sequence **416***a-f* show the quality of sequence reconstruction. The joint probability of the sequence shows how normal the input sequence to identify the temporal related anomalies as well as anomalies with large resolution. In some embodiments, the transformer generates a code level probability for each code in the sequence. By calculating the joint probability (product of the code probability) of the code sequence, the level of normalness of the code sequence may be identified, to identify the

temporal related anomalies as well as anomalies with large resolution. Normal code sequence may have a high probability to be obtained while abnormal code sequence may have a low probability. Images **412***a-c* function are used to reconstruct the original video from the sequence thus can be used for human inspection.

[0100] FIG. **5** illustrates a method for defect detection using the VBPVQAE system **100**.

[0101] For example, at **501**, an input video may be received at an encoder, and at **502**, the input video may be divided into a plurality of video patches. For example, as discussed with respect to FIG. **1**, when the input video (x) is received at the encoder (E) **120**, the input video (x) may be divided into a plurality of video patches.

[0102] The latent features from the plurality of video patches of the input video may be extracted by the encoder, and, a latent features matrix including the latent features extracted from the plurality of video patches of the input video may be determined by the encoder. For example, as discussed with respect to FIG. **1**, a latent feature from each of the plurality of patches may be extracted to determine the latent feature matrix (z) **130**, where $z \in R(h \times w \times nz)$ and $z = \in(x)$.

[0103] For example, at **505**, codes corresponding to the plurality of video patches of the input video may be selected by the encoder from a codebook including the codes. For example, as discussed with respect to FIG. **1**, after the latent feature matrix (z) **130** is determined by the encoder (E) **120**, a code for each video patch is selected (or assigned) from the codebook (Q) **140** using a look-up function q(.) by the encoder (E) **120**.

[0104] At **506**, an assigned code matrix including the codes corresponding to the plurality of video patches of the input video may be generated by the encoder. For example, as discussed with respect to FIG. **1**, by comparing the similarity measures (e.g., euclidean distance, mahalanobis distance, etc.) between the latent feature representation of an input video patch in the latent feature matrix (z) **130** and each code in the codebook (Q) **140**, the look-up function q(.) selects the nearest code as an assignment for each latent feature in the latent feature matrix (z) **130** to generate the assigned code matrix (q) **150**.

[0105] At **507**, the assigned code matrix from the encoder may be received by the decoder, and at **508**, the decoder generates a reconstructed video based on the assigned code matrix. For example, as discussed with respect to FIG. **1**, he assigned code matrix (q) **150** is passed to the decoder (G) **160** to generate a reconstructed video or a reconstructed input video (x) from the elements of the assigned code matrix (q) **150**.

[0106] A test input video may be received at the encoder and divided into a plurality of video patches. For example, as discussed with respect to FIG. **3**, an input video x at test received at the encoder may be divided into a plurality of video patches.

[0107] Latent features from the plurality of video patches of the test input video may be extracted by the encoder and each of the plurality of video patches may be encoded into a latent feature vector based on the extracted latent features. For example, as discussed with respect to FIG. **3**, after the input video x at test is divided into a plurality of video patches, an video patch of the plurality of video patches may be encoded into the latent feature vector $\hat{z}$ (shorthand for $z_{i,j}$) by the encoder (E) **120**.

[0108] Next, a code may be assigned to each of the plurality of video patches to determine assigned codes of the plurality of video patches. For example, as discussed with respect to FIG. **3**, the k^{th} code from the codebook (Q) **320** that is of the shortest distance to the latent feature vector $\hat{z}$ among all the codes in the codebook (Q) **320** (based on the similarity measure between the latent feature vector $\hat{z}$ representation of the input video patch and the k^{th} code from the codebook (Q) **320**) is assigned to the video patch (e.g., **310**(**1**), **310**(**2**), **310**(**3**) . . . , **310**(*n*)) by the encoder (E) **120**.

[0109] A patch-set including the assigned codes may be determined by the encoder. For example, as discussed with respect to FIG. **3**, with all video patches, the latent feature vector $\hat{z}$ are extracted and a patch-set **330** of the corresponding codes is determined by the encoder (E) **120**, where the patch-set **330** may include the assigned codes corresponding to each of the plurality of video patches (e.g., **310**(**1**), **310**(**2**), **310**(**3**) . . . , **310**(*n*)).

[0110] An anomaly score of each of the assigned codes of the patch-set may be determined by the encoder. For example, as discussed with respect to FIG. **3**, for each entry in the patch-set **330**, an anomaly score $s_{i,j} \in R$ may be calculated using either the probability density functions $s_{i,j} = P(\hat{z})$, or weighted distance to the k nearest-neighbors in the patch-set **330**.

[0111] The anomaly score of each of the assigned codes of the patch-set may be compared with a threshold by the encoder, and, a defect in one or more of the plurality of video patches may be determined by the encoder based on a result of the comparison. For example, as discussed with respect to FIG. **3**, during the defect detection, an input sample is determined to be defective if any of the anomaly scores $s_{i,j}$ is larger than a given threshold at index k.

[0112] Unless otherwise defined, all terms (including technical and scientific terms) used herein have the same meaning as commonly understood by one of ordinary skill in the art to which the present disclosure belongs. It will be further understood that terms, such as those defined in commonly used dictionaries, should be interpreted as having a meaning that is consistent with their meaning in the context of the relevant art and/or the present specification, and should not be interpreted in an idealized or overly formal sense, unless expressly so defined herein.

[0113] Embodiments described herein are examples only. One skilled in the art may recognize various alternative embodiments from those specifically disclosed. Those alternative embodiments are also intended to be within the scope of this disclosure. As such, the embodiments are limited only by the following claims and their equivalents

What is claimed is:

1. A method of detecting, by an encoder device, an anomaly in an input video, the method comprising:

receiving the input video;

extracting a latent feature from the input video;

selecting, based on the latent feature, a code from a codebook;

reconstructing a reconstructed video based on the selected code;

calculating a difference between the input video and the reconstructed video;

comparing the difference to a threshold; and

detecting, based on the comparison, an anomaly in the input video.

2. The method of claim 1, wherein the codebook comprises codes corresponding to a plurality of video patches of the input video, the codes in the codebook comprising the selected code, and

wherein the reconstructing the reconstructed video comprises:

determining an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video; and

generating the reconstructed video based on the assigned code matrix.

3. The method of claim 2, further comprising:

selecting the codes corresponding to the plurality of video patches of the input video using a look-up function,

wherein the look-up function, the codebook, and the assigned code matrix are stored in a memory coupled to the encoder device.

4. The method of claim 3, wherein the selecting the codes corresponding to the plurality of video patches of the input video comprises:

comparing similarity measures between a latent feature representation of a video patch of the plurality of video patches in a latent features matrix and a corresponding code in the codebook.

5. The system of claim 4, wherein the similarity measures comprise a Euclidean distance or a Mahalanobis distance between the latent feature representation of the video patch in the latent features matrix and the corresponding code in the codebook.

6. The method of claim 3, further comprising:

determining a latent features matrix comprising latent features extracted from the plurality of video patches of the input video, the latent features comprising the extracted latent feature from the input video,

selecting, by the look-up function, a nearest code as an assignment for each of the latent features in the latent features matrix; and

determining, the assigned code matrix based on the selecting the nearest code as the assignment for each of the latent features in the latent features matrix.

7. The method of claim 2, further comprising:

assigning a code from among the codes in the assigned code matrix to a video patch of the plurality of video patches of the input video based on a vector quantization of a latent feature corresponding to the video patch.

8. The method of claim 7, further comprising:

determining a vector quantization loss of the encoder device by determining a reconstruction loss that occurs during generation of the reconstructed video and a loss that occurs during the vector quantization of latent features corresponding to the plurality of video patches of the input video.

9. The method of claim 8, further comprising:

determining an adversarial training loss between the input video and the reconstructed video.

10. The method of claim 9, further comprising:

determining a total loss in the reconstructing the reconstructed video from the input video based on the vector quantization loss and the adversarial training loss.

11. The method of claim 1, further comprising:

receiving a test input video;

dividing the test input video into a plurality of video patches;

extracting latent features from the plurality of video patches of the test input video;

encoding each of the plurality of video patches into a latent feature vector based on the extracted latent features;

assigning a code to each of the plurality of video patches to determine assigned codes of the plurality of video patches;

determining a patch-set comprising the assigned codes;

determining an anomaly score of each of the assigned codes of the patch-set;

comparing the anomaly score of each of the assigned codes of the patch-set with a second threshold; and

determining a defect in one or more of the plurality of video patches based on a result of a comparison.

12. The method of claim 11, wherein a code from the codebook that is of a shortest distance to the latent feature vector of a video patch of the plurality of video patches from among the codes in the codebook is assigned to the video patch.

13. The method of claim 11, further comprising:

determining the anomaly score of each of the assigned codes of the patch-set based on a probability density function.

14. A method of detecting, by an encoder device, an anomaly in an input video, the method comprising:

extracting a latent feature from an input video;

selecting, based on the latent feature, a code from a codebook;

reconstructing a reconstructed video based on the selected code;

comparing a difference between the input video and the reconstructed video to a threshold; and

detecting, based on the comparison, an anomaly in the input video.

15. The method of claim 14, further comprising:

receiving an input video;

dividing the input video into a plurality of video patches;

selecting codes corresponding to the plurality of video patches of the input video, from the codebook comprising the codes;

determining an assigned code matrix comprising the codes corresponding to the plurality of video patches of the input video;

receiving the assigned code matrix from the encoder; and

generating the reconstructed video based on the assigned code matrix.

16. The method of claim 15, wherein the codes corresponding to the plurality of video patches of the input video are selected from the codebook using a look-up function, wherein the method further comprises:

extracting latent features from the plurality of video patches of the input video; and

determining a latent features matrix comprising the latent features extracted from the plurality of video patches of the input video.

17. The method of claim 16, wherein the look-up function is operable to select a nearest code as an assignment for each of the latent features in the latent features matrix to determine the assigned code matrix, and

wherein the encoder is operable to select a code corresponding to a video patch of the plurality of video patches of the input video from the codebook by comparing similarity measures between a latent feature

representation of the video patch in the latent features matrix and a corresponding code in the codebook.

18. The method of claim **17**, wherein a code from among the codes in the assigned code matrix is assigned to a video patch of the plurality of video patches of the input video based on a vector quantization of a latent feature corresponding to the video patch,

wherein the method further comprises:

determining a vector quantization loss of the encoder device by determining a reconstruction loss that occurs during generation of the reconstructed video and a loss that occurs during the vector quantization of latent features corresponding to the plurality of video patches of the input video, and

wherein a total loss for generating the reconstructed video from the input video comprises the vector quantization loss and an adversarial training loss.

19. The method of claim **15**, further comprising:

extracting latent features from the plurality of video patches of the input video;

encoding each of the plurality of video patches into a latent feature vector based on the extracted latent features;

assigning a code to each of the plurality of video patches to determine assigned codes of the plurality of video patches;

determining a patch-set comprising the assigned codes;

determining an anomaly score of each of the assigned codes of the patch-set;

comparing the anomaly score of each of the assigned codes of the patch-set with a second threshold; and

determining a defect in one or more of the plurality of video patches based on a result of a comparison.

20. A non-transitory computer readable storage medium operable to store instructions that, when executed by a processor included in a computing device, cause the computing device to:

receive an input video;

extract a latent feature from the input video;

select, based on the latent feature, a code from a codebook;

reconstruct a reconstructed video based on the selected code;

calculate a difference between the input video and the reconstructed video;

compare the difference to a threshold; and

detect, based on the comparison, an anomaly in the input video.

* * * * *