



(51) International Patent Classification:

G06T 11/60 (2006.01) G06F 3/14 (2006.01)
G06T 5/50 (2006.01) G06N 3/08 (2006.01)

(21) International Application Number:

PCT/KR2021/000224

(22) International Filing Date:

08 January 2021 (08.01.2021)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2020133033 07 October 2020 (07.10.2020) RU

(71) Applicant: SAMSUNG ELECTRONICS CO., LTD. [KR/KR]; 129, Samsung-ro, Yeongtong-gu, Suwon-si, Gyeonggi-do 16677 (KR).

(72) Inventors: SUVOROV, Roman Evgenievich; office #1500, 12 Dvintsev street, korpus 1, Moscow, 127018 (RU). LOGACHEVA, Elizaveta Mikhailovna; office #1500, 12 Dvintsev street, korpus 1, Moscow, 127018 (RU). LEM-PITSKY, Victor Sergeevich; office #1500, 12 Dvintsev street, korpus 1, Moscow, 127018 (RU). MASHIKHIN, Anton Evgenievich; office #1500, 12 Dvintsev street, korpus 1, Moscow, 127018 (RU). KHOMENKO, Oleg Igore-

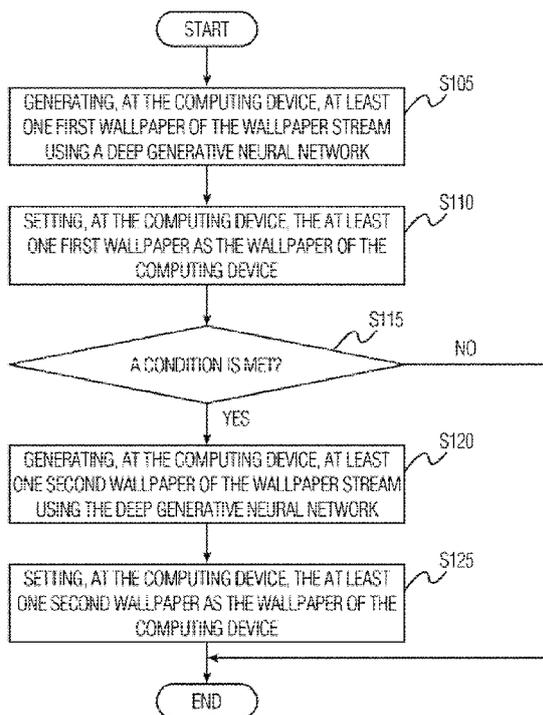
vich; office #1500, 12 Dvintsev street, korpus 1, Moscow, 127018 (RU).

(74) Agent: KIM, Tae-hun et al.; 9th Floor, Shinduk Bldg., 343, Gangnam-daero, Seocho-gu, Seoul 06626 (KR).

(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, IT, 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: METHOD OF ON-DEVICE GENERATION AND SUPPLYING WALLPAPER STREAM AND COMPUTING DEVICE IMPLEMENTING THE SAME



(57) Abstract: A method of on-device generation and supplying a computing device with wallpaper stream is disclosed. The method comprise: generating (S105), at the computing device, at least one first wallpaper of the wallpaper stream using a deep generative neural network, wherein the deep generative neural network is trained on a collection of high-quality images/videos and uploaded to the computing device in advance; and setting (S110), at the computing device, the at least one first wallpaper as the wallpaper of the computing device.



WO 2022/075533 A1

Published:

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

Description

Title of Invention: METHOD OF ON-DEVICE GENERATION AND SUPPLYING WALLPAPER STREAM AND COMPUTING DEVICE IMPLEMENTING THE SAME

Technical Field

- [1] The present invention relates to the field of artificial intelligence in general, and specifically, to a method of generating and supplying a wallpaper stream on a computing device using a deep generative neural network and to the computing device implementing the method.

Background Art

- [2] Wallpapers are a big part of user experience for a variety of devices, including smartphones, smart TVs, laptops and so on. Currently, to obtain new aesthetically pleasant wallpapers regularly users may subscribe to online updates, i.e. services (like *Unsplash*) that regularly send new wallpapers to the user device via Internet connection. Disadvantages of such an approach comprise at least the need of Internet connection, and traffic and bandwidth consumption.

Disclosure of Invention

Technical Problem

- [3] Generative neural networks are now capable of synthesizing highly realistic looking 2D images, 3D images and videos. Such networks can therefore be trained to generate realistic aesthetically pleasant wallpaper images. Once trained, many of these models can generate infinite number of highly diverse wallpapers (the most popular class of these models, namely generative adversarial networks) by taking a random high-dimensional vector as an input and generating a peculiar image for such vector. Plugging in a new vector would result in a substantially different image.

Solution to Problem

- [4] Proposed herein is the alternative invention that allows regularly update the wallpapers of the user device without using the Internet. Provided according to the first aspect of the present disclosure is a method of on-device generation and supplying a computing device with wallpaper stream, the method including: generating, at the computing device, at least one first wallpaper of the wallpaper stream using a deep generative neural network, wherein the deep generative neural network is trained on a collection of high-quality images/videos and uploaded to the computing device in advance; and setting, at the computing device, the at least one first wallpaper as the wallpaper of the computing device. Since the plausible wallpaper is synthesized on the

computing device itself, i.e. not downloaded from Internet, the above disadvantages are eliminated and the claimed invention makes it possible to reduce/avoid traffic and bandwidth consumption that was needed in the prior art for downloading.

Advantageous Effects of Invention

- [5] Provided according to the second aspect of the present disclosure is a computing device including a processor and a storage device storing the trained deep generative neural network for on-device generation and supplying a wallpaper stream by performing the method according to the first aspect, when the trained deep generative neural network is executed by the processor.

Brief Description of Drawings

- [6] Figure 1 illustrates a flowchart of a method of on-device generation and supplying a computing device with wallpaper stream according to an embodiment of the invention disclosed herein.
- [7] Figure 2 illustrates a flowchart of a method of on-device generation and supplying a computing device with wallpaper stream according to another embodiment of the invention disclosed herein.
- [8] Figure 3 illustrates a structural diagram of a computing device according to an embodiment of the invention disclosed herein.
- [9] In the following description, unless otherwise described, the same reference numerals are used for the same elements when they are depicted in different drawings, and overlapping description thereof may be omitted.

Best Mode for Carrying out the Invention

- [10] Figure 1 illustrates a flowchart of a method of on-device generation and supplying a computing device with wallpaper stream according to an embodiment of the invention disclosed herein. The method comprise: generating S105, at the computing device, at least one first wallpaper of the wallpaper stream using a deep generative neural network, wherein the deep generative neural network is trained on a collection of high-quality images/videos and uploaded to the computing device in advance. Implied by said generating is artificial synthesis of a wallpaper with the deep generative neural network. Depending on a type of content in the collection of high-quality images/videos, which was used during the training phase of the deep generative neural network, the deep generative neural network, when trained with such a collection, is configured to generate wallpapers of similar content type. For example, when the collection of high-quality images includes landscape images, the deep generative neural network, when trained with such a collection, will be configured to generate landscape wallpapers and so on. However, the present invention should not be limited with landscape wallpapers, because a collection of high-quality images/videos of any

other type of content may be used at the training phase of the deep generative neural network. Furthermore, if a collection of high-quality videos was used during the training phase of the deep generative neural network, the deep generative neural network, when trained with such a collection, will be configured to generate video wallpapers of a content type corresponding to a content type of the training collection of high-quality videos. Implied by the wallpaper stream is one or more images that could be static, dynamic or interactive and/or one or more videos. When the deep generative neural network is trained, the deep generative neural network including weighting factors and other parameters may be uploaded to the computing device in advance, i.e. before the inference phase. The deep generative neural network may be stored in a storage device such as a memory of the computing device. When the at least one first wallpaper is generated by the deep generative neural network, the method comprises the step of setting S110, at the computing device, the at least one first wallpaper as the wallpaper of the computing device. The generated wallpaper may be for any kind of user interface, for example, the generated wallpaper may be the wallpaper for the main desktop, for the lock screen, for the empty page of a browser and so on without limitation. The generated wallpaper may be used as a screensaver of the computing device.

- [11] As illustrated on Figure 1 the method further comprises the step of determining S115, whether a condition is met or not. This conditioning is used to determine whether it is necessary to update the first wallpaper with a second wallpaper. The condition includes, but is not limited to, one or more of the following: (i) a user input is received at the computing device, wherein the user input may represent whether a wallpaper currently set as the wallpaper of the computing device is disliked by a user of the computing device or not; (ii) a preconfigured period of time is elapsed; (iii) GPS location of the computing device is changed, wherein the GPS location of the computing device may be registered by a GPS unit comprised in the computing device. Based on dislikes the method may be adapted to generate another wallpaper if the currently set wallpaper is disliked by the user. From such like/dislike information, the system can learn to generate wallpapers that the user will like. If it is determined that the condition is met (i.e. YES at step S115) then the method updates the wallpaper currently set as the wallpaper of the computing device by performing the following steps: generating S120, at the computing device, at least one second wallpaper of the wallpaper stream using the deep generative neural network, setting S125, at the computing device, the at least one second wallpaper as the wallpaper of the computing device. The described wallpaper update may be performed automatically in the background. As an example, the user can enable a feature which generates a new wallpaper every morning. The terms *<<first>>* and *<<second>>* when applied to the

term «*wallpaper*» are used to differentiate between different wallpapers and should not be construed as the terms representing any ordinal relationship between said wallpapers or between the steps of the method. The at least one second wallpaper differs from the at least one first wallpaper. In an alternative embodiment (not illustrated) of the method, the step of determining S115, whether a condition is met or not, may be performed before generating and setting each subsequent wallpaper, including the case where the step of determining S115 is performed before the steps S105 and S110 explained above. Thus, it should be clear that the reference numerals are used just to simplify the illustration and should not be construed as representing any ordinal relationship between the steps of the method.

[12] Figure 2 illustrates a flowchart of a method of on-device generation and supplying a computing device with wallpaper stream according to another embodiment of the invention disclosed herein. The method embodiment illustrated on Figure 2 differs from the method embodiment illustrated on Figure 1 in that it further comprises the steps of individualizing S95 the deep generative neural network for a user of the computing device using a random input as a parameter of the deep generative neural network thereby ensuring that the deep generative neural network is configured to generate unique wallpaper for the user of the computing device, and customizing S100 the deep generative neural network for the user of the computing device thereby ensuring that a wallpaper generated by the deep generative neural network is customized for the user. The steps S105-S125 illustrated on Figure 2 may be similar to the S105-S125 illustrated on Figure 1 except that the at least one first wallpaper and the at least one second wallpaper generated and set by the embodiment of Figure 2 are individualized and customized. Therefore, every user gets her/his own unique wallpaper every time.

[13] The customization S100 may be based, but without the limitation, on one or more of the following customization parameters: one or more user preferences, one or more user inputs, one or more settings of the computing device, current time of the day, a current season of the year, current GPS location of the computing device, content of the user gallery currently stored on the computing device, content of browser history currently stored on the computing device, current weather and weather forecast, location and colors of icons and widgets on the device screen. Current location and colors of icons and widgets on the device screen may be detected and used in the disclosed method to synthesize such a wallpaper that is not mixed with the icons and widgets. In this case, current location and colors of icons and widgets on the device screen may be input to the deep generative neural network before generating/synthesizing a wallpaper as the corresponding parameters to effect an output of the deep generative neural network correspondingly. The method may further include (not il-

illustrated) the steps of analyzing the user gallery content, inferring that the user is fond of mountain photography, and adapting the deep generative neural network to generate more images/videos/interactive wallpapers of mountains. Despite the fact that Figure 2 shows that the steps S95 and S100 are performed before the steps S105 and S110, it should not be construed as a limitation, because, if necessary, the steps may be ordered differently, for example, one or both of the steps S95 and S100 may be performed before generating and setting each subsequent wallpaper, including the case where one or both of the steps S95 and S100 is performed before the steps S120 and S125 explained above. Thus, it should be clear that the reference numerals are used just to simplify the illustration and should not be construed as representing any ordinal relationship between the steps of the method.

[14] The deep generative neural network is trained using adversarial learning process alongside with one or more discriminator networks. The training is performed on a high-end computer or a computational cluster on a large dataset of wallpaper-quality images and/or videos. The deep generative neural network may have one or more of the following variables: vectorial variables, latent variables shaped as a two-dimensional matrix or a set of two-dimensional matrices. In an embodiment, latent variables may be drawn from unit normal distributions. The customization may be accomplished by a separate encoder network that is trained to map the customization parameters to the parameters of the latent space distributions such as the mean and the covariance of the Gaussian normal distribution, from which the latent variables are drawn for the deep generative neural network. When user-sensitive information such as, for example, one or more user preferences, one or more user inputs, one or more settings of the computing device, current GPS location of the computing device, content of the user gallery currently stored on the computing device, content of browser history currently stored on the computing device is processed as the customization parameter(s), it should be clear that such user-sensitive information is not compromised, because the entire processing of said user-sensitive information is carried out by a processor of the user's computing device and by the separate encoder network and the deep generative neural network, which are stored in the storage device of the user's computing device. In other words, advantageously, the user-sensitive information does not leave the computing device for processing.

[15] In an alternative embodiment (not illustrated) of the method the step of generating S105, S120 at least one wallpaper further comprises the steps of synthesizing an image and restyling the image, and the step of setting S110, S125 the at least one wallpaper as the wallpaper of the computing device further comprises the step of setting the restyled image as the at least one first wallpaper. In still another embodiment (not illustrated) of the method the step of generating S105, S120 at least one wallpaper further comprises

the steps of synthesizing an image and animating the image, and the step of setting S110, S125 the at least one wallpaper as the wallpaper of the computing device further comprises the step of setting the animated image as the at least one first wallpaper. Because there are no constraints on the bandwidth and acquiring the content is essentially free, dynamic wallpapers (videos at high resolution) can be generated with an appropriate model. Apart from dynamic wallpapers, interactive wallpapers can be generated. A smartphone wallpaper can change appearance based on user swipes, phone tilts or some interface events. As an example, swiping to a different tab in Android screen can change something in an image (e.g. move clouds in the image in the swipe direction). According to another alternative embodiment of the disclosed method the deep generative neural network may be adapted to generate not only realistic and plausible images but also hyper-realistic images that can have e.g. exaggerated features, such as extremely saturated sunset colors, exaggerated geometric proportions of objects (trees, buildings) etc. Some users may like and prefer such hyper-realistic wallpapers. Most generative models (e.g. adversarially trained generative neural networks) can allow each user to set her/his own preferred tradeoff between realism and hyperrealism. In still another embodiment of the method the step of generating S105, S120 at least one wallpaper further comprises the step of applying super resolution to the synthesized image, and the step of setting S110, S125 the at least one wallpaper as the wallpaper of the computing device further comprises the step of setting the image having super-resolution as the at least one first wallpaper. Particular techniques of restyling an image, applying super-resolution or hyper-resolution to an image are known in the art.

- [16] Figure 3 illustrates a structural diagram of a computing device 200 according to an embodiment of the invention disclosed herein. The computing device (200) comprises a processor 205 and a storage device 210. The processor 205 is configured to perform processing and computing tasks related to the operation of the computing device and to the operation on the method disclosed herein. The storage device stores the trained deep generative neural network 210.1 for on-device generation and supplying a wallpaper stream by performing the disclosed, when the trained deep generative neural network 210.1 is executed by the processor 205. The storage device 210 may also store processor-executable instructions to cause the processor to perform any one or more of the above described method steps. The processor 205 and the storage device 210 may be operatively interconnected. The processor 205 and the storage device 210 may be also connected with other components (not illustrated) of the computing device. The other component may include, but without the limitation, one or more of a display, a touchscreen, a keyboard, a keypad, a communication unit, a speaker, a microphone, a camera, Bluetooth-unit, NFC (Near-Field Communication) unit, RF(Radio-Frequency)

unit, GPS unit, I/O means, as well as necessary wirings and interconnections and so on. The processor 205 may be implemented, but without the limitation, a general-purpose processor, an application-specific integrated circuit (ASIC), a user-programmable gate array (FPGA), or a system-on-chip (SoC). The storage device 210 may include, but without the limitation, RAM, ROM and so on. Thus, the computing device may be, but without the limitation, the user's computing device being one of a smartphone, a tablet, a laptop, a notebook, smart TV, in-vehicle infotainment system and so on.

[17] Any piece of the following information should not be construed as a limitation on the present disclosure. Instead, the following information is provided below to enable the skilled person to practice the embodiments disclosed herein and to prove that the disclosure is sufficient. Any particular values of any parameters specified below are to be construed non-limiting.

[18] Model architecture. The architecture of the model may be based on StyleGAN. The model outputs images of resolution 256 X 256 (or 512 X 512) and has four sets of latent variables:

[19] - a vector $z^{st} \in R^{D^{st}}$, which encodes colors and the general scene layout;

[20] - a vector $z^{dyn} \in R^{D^{dyn}}$, which encodes global lighting (e.g. time of day);

[21] - a set S^{st} of square matrices $S_1^{st} \in R^{4 \times 4}, \dots, S_N^{st} \in R^{2^{N+1} \times 2^{N+1}}$, which encode

shapes and details of static objects at $N = 7$ different resolutions between 4 X 4 and 256 X 256 ($N = 8$ for 512 X 512);

[22] - a set S^{dyn} of square matrices $S_1^{dyn} \in R^{4 \times 4}, \dots, S_N^{dyn} \in R^{2^{N+1} \times 2^{N+1}}$, which encode

shapes and details of dynamic objects at the corresponding resolutions.

[23] The generator has two components: the multilayer perceptron M and the convolutional generator G. The perceptron M takes the concatenated vector

$z = [z^{st}, z^{dyn}] \in R^{512}$ and transforms it to the style vector $w \in R^{512}$. The convolutional generator G has $N = 7$ (or 8) blocks. Within each block, a convolution is

followed by two elementwise additions of two tensors obtained from S_n^{st} and S_n^{dyn} by

a learnable per-channel scaling. Finally, the AdaIN transform is applied using per-

channel scales and biases obtained from w using learnable linear transform. Within

each block, this sequence of steps is repeated twice followed by upsampling and convolution layers.

[24] Below, the following set of input latent variables will be referred to

[25] $\{z^{st}, z^{dyn}, S_1^{st}, \dots, S_N^{st}, S_1^{dyn}, \dots, S_N^{dyn}\}$

[26] as original inputs (or original latents). As in StyleGAN, the convolutional generator may use separate w vectors at each of the resolution (style mixing). The set of all

style vectors will be referred to as $W = \{w^1, \dots, w^N\}$. Finally, the set of all spatial random inputs of the generator will be denoted as

$$S = \{S^{st}, S^{dyn}\} = \{S_1^{st}, \dots, S_N^{st}, S_1^{dyn}, \dots, S_N^{dyn}\}.$$

- [27] Learning the model. The model is trained from two sources of data, the dataset of static scenery images \mathcal{I} and the dataset of timelapse scenery videos \mathcal{V} . It is relatively easy to collect a large static dataset, while with authors best efforts a few hundreds of videos that do not cover all the diversity of landscapes were collected. Thus, both sources of data may be utilized in order to build a model having improved performance. To do that, the proposed generative model (the deep generative neural network) is trained in an adversarial way with two different discriminators.
- [28] The static discriminator D_{st} has the same architecture and design choices as in StyleGAN. It observes images \mathcal{I} from as real, while the fake samples are generated by the model. The pairwise discriminator D_{dyn} looks at pairs of images. It duplicates the architecture of D_{st} except first convolutional block that is applied separately to each frame. A real pair of images is obtained by sampling a video from \mathcal{V} , and then sampling two random frames (arbitrary far from each other) from it. A fake pair is obtained by sampling common static latents z^{st} and S^{st} , and then individual dynamic latents $z^{dyn,1}$, $z^{dyn,2}$, and $S^{dyn,1}$, $S^{dyn,2}$. The two images are then obtained as $G(M(z^{st}, z^{dyn,1}), S^{st}, S^{dyn,1})$ and $G(M(z^{st}, z^{dyn,2}), S^{st}, S^{dyn,2})$. All samples are drawn from unit normal distributions.
- [29] The model is trained within GAN approach with non-saturating loss with R1 regularization. During each update of the generator, a batch of fake images to which the static discriminator is applied is sampled or a batch of image pairs to which the pairwise discriminator is applied is sampled. The proportions of the static discriminator and the pairwise discriminator are selected from 0.5/0.5 to 0.9/0.1 respectively over each resolution transition phase and then kept fixed at 0.1. This helps the generator to learn disentangle static and dynamic latents early for each resolution and prevents the pairwise generator from overfitting to relatively small video dataset used for training.
- [30] During learning, the objective of the pairwise discriminator is to focus on the inconsistencies within each pair, and the objective of the static discriminator is to focus on visual quality. Furthermore, since the pairwise discriminator only sees real frames sampled from a limited number of videos, it may prone overfit to this limited set and effectively stop contributing to the learning process (while the static discriminator, which observes more diverse set of scenes, keeps improving the diversity of the model). It turns out both problems (focus on image quality rather than pairwise con-

sistency, overfitting to limited diversity of videos) can be solved with a simple method. This method consists in that the fake set of frames are augmented with pairs of crops taken from same video frame, but from different locations. Since these crops have the same visual quality as the images in real frames, and since they come from the same videos as images within real pairs, the pairwise discriminator effectively stops paying attention to image quality, cannot simply overfit to the statistics of scenes in the video dataset, and has to focus on finding pairwise inconsistencies within fake pairs. This crop sampling method may be used to improve the quality of the model significantly.

[31] Sampling videos from the model. The model does not attempt to learn full temporal dynamics of videos, and instead focuses on pairwise consistency of frames that are generated when the dynamic latent variables are resampled. In particular, the pairwise discriminator in the model does not sample real frames sequentially. The sampling procedure for fake pairs does not try to generate adjacent frames either. One of the reasons why the described learning phase does not attempt to learn continuity, is because the training dataset contains videos of widely-varying temporal rates, making the notion of temporal adjacency for a pair of frames effectively meaningless.

[32] Because of this, the disclosed generation process is agnostic to a model of motion. The generator is forced to produce plausible frames regardless of S^{dyn} and Z^{dyn} changes. In the experiments it was found that a simple model of motion described below is enough to produce compelling videos. Specifically, to sample a video, it is possible to sample a single static vector Z^{st} from the unit normal distribution and then interpolate the dynamic latent vector between two unit normally distributed samples $Z^{dyn,1}$ and $Z^{dyn,2}$. For the spatial maps, again it is possible to sample S^{st} and $S^{dyn,1}$ from a unit normal distribution and then warp the S^{dyn} tensor continuously using a homography transform parameterized by displacements of two upper corners and two points at the horizon. The direction of the homography is sampled randomly, speed was chosen to match the average speed of clouds in the training dataset. The homography is flipped vertically for positions below the horizon to mimic the reflection process. To obtain $S^{dyn,i}$, it is possible to make a composition of $i-1$ identical transforms and then apply it to $S^{dyn,1}$. As the latent variables are interpolated/warped, the latent variables are passed through the trained model to obtain the smooth videos. It is necessary to note that the described model requires no image-specific user input.

[33] Animating Real Scenery Images with the Model. *Inference.* To animate a given scenery image I , a set of latent variables that produce such an image within the generator is inferred. Extended latents W and S are looked for, so that $G(W,S) \approx I$.

After that, the same procedure as described above to animate the given image may be applied.

- [34] The latent space of the generator is highly redundant, and to obtain good animation, it has to be ensured that the latent variables come roughly from the same distribution as during the training of the model (most importantly, \mathcal{W} should belong to the output manifold of \mathcal{M}). Without such prior, the latent variables that generate good reconstruction might still result in implausible animation (or lack of it). Therefore inference may be performed using the following three-step procedure:
- [35] 1. Step 1: predicting a set of style vectors \mathcal{W}' using a feedforward encoder network E . The encoder has ResNet-152 architecture and is trained on 200000 synthetic images with mean absolute error loss. \mathcal{W} is predicted by two-layer perceptron with ReLU from the concatenation of features from several levels of ResNet, aggregated by global average pooling.
- [36] 2. Step 2: starting from \mathcal{W}' and zero \mathcal{S} , all latents are optimized to improve reconstruction error. In addition, the deviation of \mathcal{W} from the predicted \mathcal{W}' (with coefficient 0.01) and the deviation of \mathcal{S} from zero (by reducing learning rate) are penalized. Optimization is performed for up to 500 steps with Adam and large initial learning rate (0.1), which is halved each time the loss does not improve for 20 iterations. A variant of the method, which was evaluated separately, uses a binary segmentation mask obtained by ADE20k-pretrained segmentation network. The mask identifies dynamic (sky+water) and remaining (static) parts of the scene. In this variant, \mathcal{S}^{st} (respectively \mathcal{S}^{dyn}) are kept at zero for dynamic (respectively, static) parts of the image.
- [37] 3. Step 3: freezing latents and fine-tuning the weights of G to further drive down the reconstruction error. This step is needed since even after optimization, the gap between the reconstruction and the input image remains. During this fine-tuning, the combination of the per-pixel mean absolute error and the perceptual loss are minimized, with much larger (10X) weight for the latter. 500 steps with ADAM and $lr = 0,001$ are performed.
- [38] Lighting manipulation. During training of the model, \mathcal{M} is used to map z to w . z^{dyn} is resampled in order to take into account variations of lighting, weather changes, etc. and to have z^{st} describe only static attributes (land, buildings, horizon shape, etc.). To change lighting in a real image, one has to change z^{dyn} and then use MLP to obtain new styles \mathcal{W} . The described inference procedure, however, outputs \mathcal{W} and it has been found that it is very difficult to invert \mathcal{M} and obtain $z = \mathcal{M}^{-1}(w)$.

[39] To tackle this problem, a separate neural network, \mathcal{A} , is trained to approximate local dynamics of \mathcal{M} . Let $w_a = M(z_a^{st}, z_a^{dyn})$ and $w_b = M(z_b^{st}, z_b^{dyn})$. \mathcal{A} is optimized as follows: $A(w_a, z_b^{dyn}, c) \approx M(z_a^{st}, z_a^{dyn} \sqrt{1-c} + z_b^{dyn} \sqrt{c})$, where $c \sim \text{Uniform}(0, 1)$ is coefficient of interpolation between w_a and w_b . Thus, $c = 0$ corresponds to z_a^{dyn} , so $A(w_a, z_b^{dyn}, 0) \approx w_a$; $c = 1$ corresponds to z_b^{dyn} , so $A(w_a, z_b^{dyn}, 1) \approx w_b$.

[40] This is implemented by the combination of L1-loss $L_{Abs}^A = |w_b - A(\cdot)|$ and relative direction loss $L_{Rel}^A = 1 - \cos(w_b - w_a, A(\cdot) - w_a)$. The total optimization criterion is $L^A = L_{Abs}^A + 0.1 L_{Rel}^A$. \mathcal{A} is trained with ADAM until convergence. At test time, the network \mathcal{A} makes it possible to sample a random target z_b^{dyn} and update W towards it by increasing the interpolation coefficient c as the animation progresses.

[41] Super Resolution (SR). As the models are trained at medium resolution (e.g. 256X256), it becomes possible to bring fine details from a given image that is subject to animation through a separate super-resolution procedure. The main idea of the super resolution approach is to borrow as much as possible from the original high-res image (which is downsampled for animation via \mathcal{G}). To achieve that, the animation is super-resolved and blended with the original image using a standard image super resolution approach. It is possible to use ESRGANx4 trained on a dedicated dataset that is created as follows. To obtain the (hi-res, low-res) pair, a frame I is taken from the video dataset as a hi-res image, the frame is then downsampled and the first two steps of inference are run and an (imperfect) low-res image is obtained. Thus, the network is trained on a more complex task than super resolution.

[42] After obtaining the super-resolved video, dynamic parts (sky and water) are transferred from it to the final result. The static parts are obtained by running the guided filter on the super-resolved frames while using the input high-res image as a guide. Such procedure effectively transfers high-res details from the input, while retaining the lighting change induced by lighting manipulation.

[43] At least one of the plurality of modules, units, components, steps, sub-steps may be implemented through an AI model. A function associated with AI may be performed through the non-volatile memory, the volatile memory, and the processor. The processor may include one or a plurality of processors. At this time, one or a plurality of processors may be a general-purpose processor, such as a central processing unit (CPU), an application processor (AP), or the like, a graphics-only processing unit such as a graphics processing unit (GPU), a visual processing unit (VPU), and/or an AI-dedicated processor such as a neural processing unit (NPU). The one or a plurality of

processors control the processing of the input data in accordance with a predefined operating rule or artificial intelligence (AI) model stored in the non-volatile memory and the volatile memory. The predefined operating rule or artificial intelligence model is provided through training or learning. Here, being provided through learning means that, by applying a learning algorithm to a plurality of learning data, a predefined operating rule or AI model of a desired characteristic is made. The learning may be performed in a device itself in which AI according to an embodiment is performed, and/or may be implemented through a separate server/system.

- [44] The AI model may consist of a plurality of neural network layers. Each layer has a plurality of weight values, and performs a layer operation through calculation of a previous layer and an operation of a plurality of weights. Examples of neural networks include, but are not limited to, convolutional neural network (CNN), deep neural network (DNN), recurrent neural network (RNN), restricted Boltzmann Machine (RBM), deep belief network (DBN), bidirectional recurrent deep neural network (BRDNN), generative adversarial networks (GAN), and deep Q-networks. The learning algorithm is a method for training a predetermined target device (for example, a robot) using a plurality of learning data to cause, allow, or control the target device to make a determination or prediction. Examples of learning algorithms include, but are not limited to, supervised learning, unsupervised learning, semi-supervised learning, or reinforcement learning.
- [45] It should be expressly understood that not all technical effects mentioned herein need to be enjoyed in each and every embodiment of the present technology. For example, embodiments of the present technology may be implemented without the user enjoying some of these technical effects, while other embodiments may be implemented with the user enjoying other technical effects or none at all.
- [46] Modifications and improvements to the above-described implementations of the present technology may become apparent to those skilled in the art. The foregoing description is intended to be exemplary rather than limiting. The scope of the present technology is therefore intended to be limited solely by the scope of the appended claims.
- [47] While the above-described implementations have been described and shown with reference to particular steps performed in a particular order, it will be understood that these steps may be combined, sub-divided, or reordered without departing from the teachings of the present technology. Accordingly, the order and grouping of the steps is not a limitation of the present technology.

Claims

- [Claim 1] A method of on-device generation and supplying a computing device with wallpaper stream, comprising:
generating (S105), at the computing device, at least one first wallpaper of the wallpaper stream using a deep generative neural network, wherein the deep generative neural network is trained on a collection of high-quality images/videos and uploaded to the computing device in advance; and
setting (S110), at the computing device, the at least one first wallpaper as the wallpaper of the computing device.
- [Claim 2] The method of claim 1, further comprising the following steps performed upon a condition is met (S115):
generating (S120), at the computing device, at least one second wallpaper of the wallpaper stream using the deep generative neural network, wherein the at least one second wallpaper differs from the at least one first wallpaper;
setting (S125), at the computing device, the at least one second wallpaper as the wallpaper of the computing device.
- [Claim 3] The method of claim 2, wherein the condition includes one or more of the following conditions:
a user input is received at the computing device, wherein the user input represents whether a wallpaper currently set as the wallpaper of the computing device is disliked by a user of the computing device or not;
a preconfigured period of time is elapsed;
GPS location of the computing device is changed, wherein the GPS location of the computing device is registerable by a GPS unit comprised in the computing device.
- [Claim 4] The method of claim 1, further comprising one or both of the following steps:
individualizing (S95) the deep generative neural network for a user of the computing device using a random input as a parameter of the deep generative neural network thereby ensuring that the deep generative neural network is configured to generate unique wallpaper for the user of the computing device;
customizing (S100) the deep generative neural network for the user of the computing device thereby ensuring that a wallpaper generated by the deep generative neural network is customized for the user, the cus-

tomization is based on one or more of the following customization parameters: one or more user preferences, one or more user inputs, one or more settings of the computing device, current time of the day, a current season of the year, current GPS location of the computing device, content of the user gallery currently stored on the computing device, content of browser history currently stored on the computing device, current weather and weather forecast, location and colors of icons and widgets on the device screen.

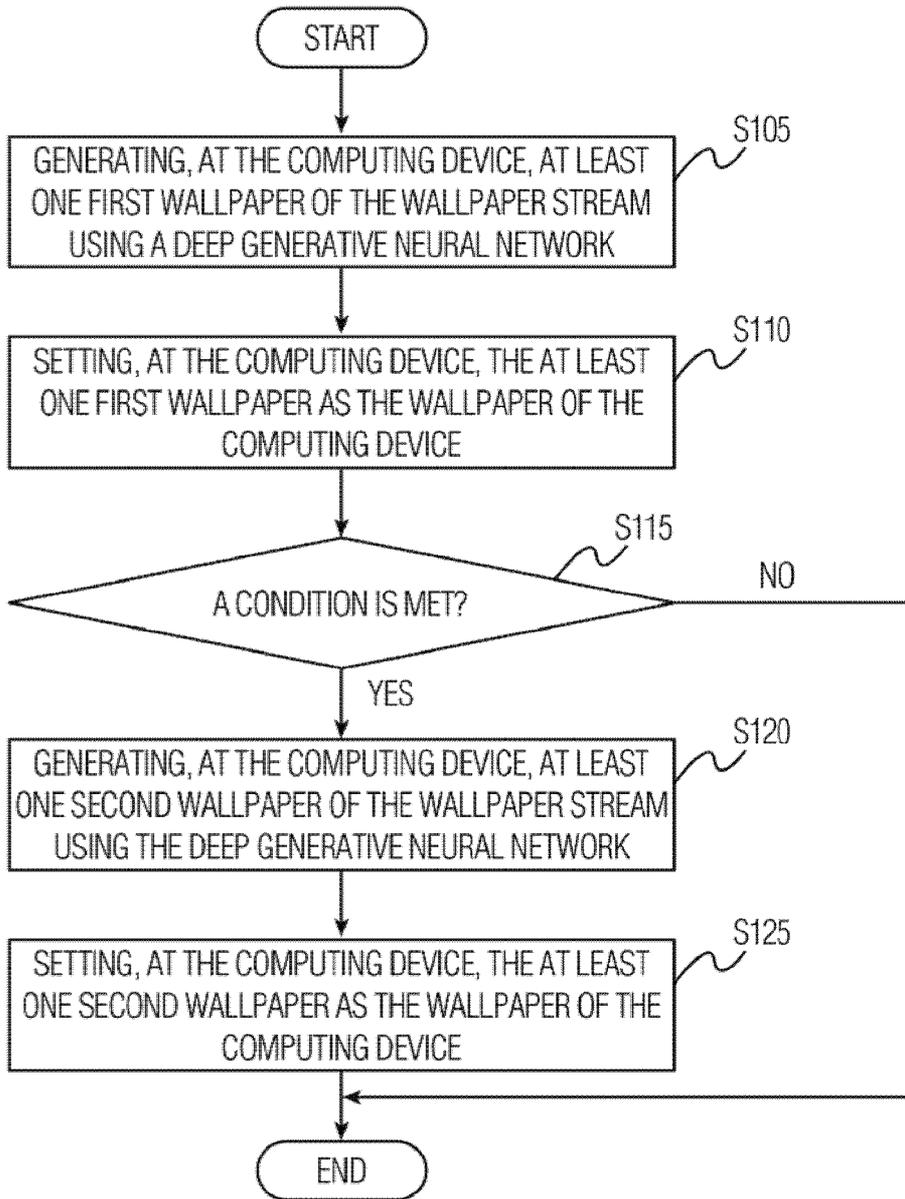
- [Claim 5] The method of claim 1 or 2, wherein the wallpaper generated by the deep generative neural network is static, dynamic or interactive.
- [Claim 6] The method of claim 1, wherein the deep generative neural network has one or more of the following variables: vectorial variables, latent variables shaped as a two-dimensional matrix or a set of two-dimensional matrices.
- [Claim 7] The method of claim 6, wherein latent variables are drawn from unit normal distributions.
- [Claim 8] The method of any one of claims 4 or 6, wherein the customization is accomplished by a separate encoder network that is trained to map the customization parameters to the parameters of the latent space distributions such as the mean and the covariance of the Gaussian normal distribution, from which latent variables are drawn.
- [Claim 9] The method of claim 1 or 2, wherein the step of generating at least one wallpaper further comprises the steps of synthesizing an image and restyling the image, and the step of setting the at least one wallpaper as the wallpaper of the computing device further comprises the step of setting the restyled image as the at least one first wallpaper.
- [Claim 10] The method of claim 1 or 2, wherein the step of generating at least one wallpaper further comprises the steps of synthesizing an image and animating the image, and the step of setting the at least one wallpaper as the wallpaper of the computing device further comprises the step of setting the animated image as the at least one first wallpaper.
- [Claim 11] The method of claim 6, wherein the deep generative neural network is trained using adversarial learning process alongside with one or more discriminator networks.
- [Claim 12] A computing device (200) comprising a processor (205) and a storage device (210) storing the trained deep generative neural network (210.1)

for on-device generation and supplying a wallpaper stream by performing the method of any one of claims 1-11, when the trained deep generative neural network (210.1) is executed by the processor (205).

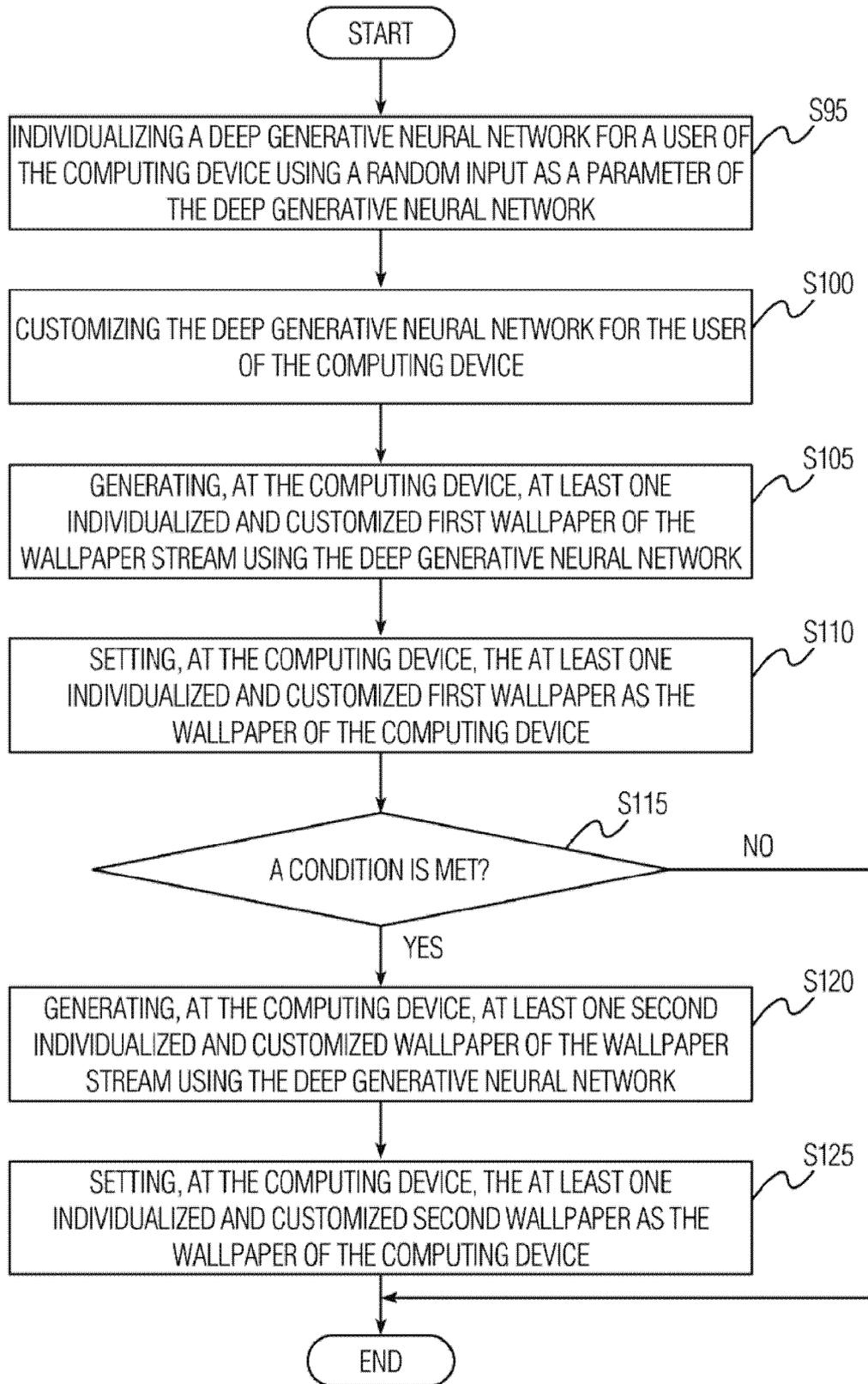
[Claim 13]

The computing device (200) of claim 12, wherein the computing device (200) comprises the user-computing device being one of the following: a smartphone, a tablet, a laptop, a notebook, smart TV.

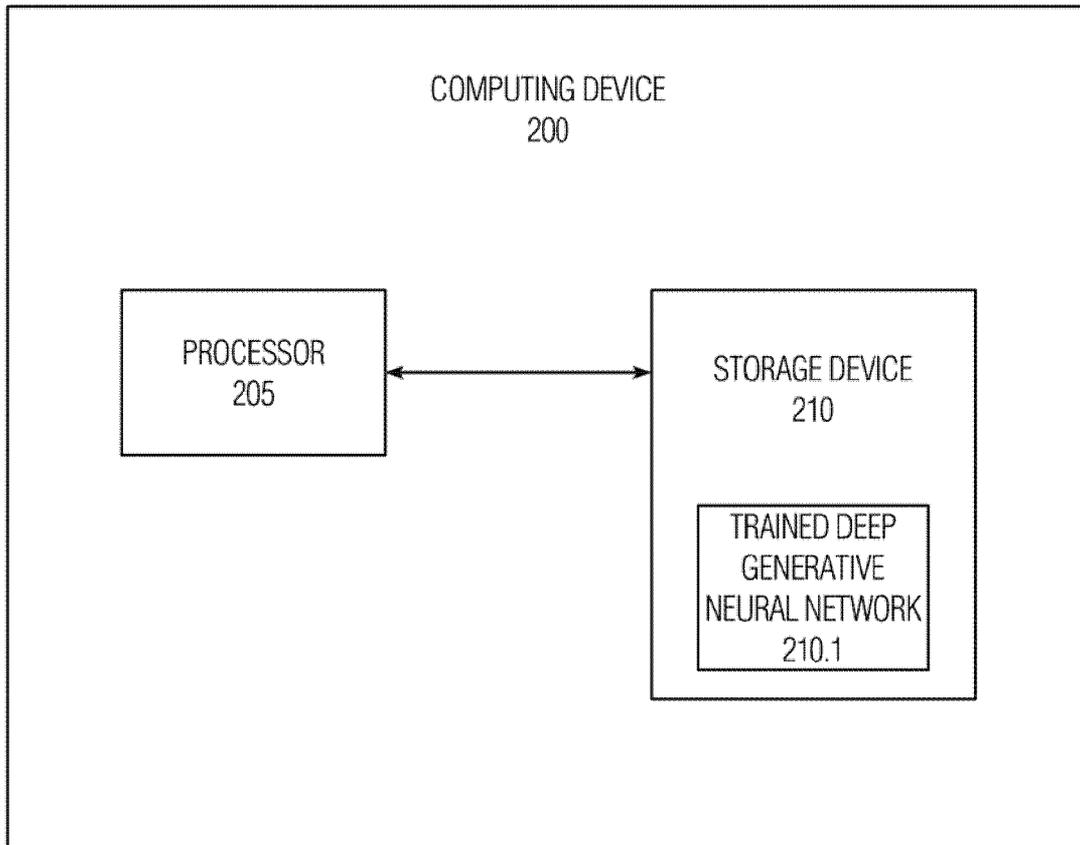
[Fig. 1]



[Fig. 2]



[Fig. 3]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/KR2021/000224

A. CLASSIFICATION OF SUBJECT MATTER		
G06T 11/60(2006.01)i; G06T 5/50(2006.01)i; G06F 3/14(2006.01)i; G06N 3/08(2006.01)i		
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) G06T 11/60(2006.01); A61M 21/02(2006.01); G06F 17/30(2006.01); G06Q 30/02(2012.01); G06T 3/00(2006.01); G06T 5/00(2006.01); H04B 1/3827(2014.01); H04M 1/725(2006.01); H04N 21/466(2011.01)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Korean utility models and applications for utility models Japanese utility models and applications for utility models		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) eKOMPASS(KIPO internal) & keywords: wallpaper, stream, neural network, image, video, train, condition		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2018-0012330 A1 (FYUSION, INC) 11 January 2018 (2018-01-11) paragraphs [0012], [0061], [0073], [0295]-[0323]; and claim 1	1-13
Y	US 2019-0251721 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 15 August 2019 (2019-08-15) paragraphs [0002], [0013], [0024]-[0025], [0040]; claims 1-2; and figure 2	1-13
A	US 2018-0150203 A1 (AT&T MOBILITY II LLC) 31 May 2018 (2018-05-31) paragraph [0037]; and figures 3-9	1-13
A	US 2018-0075490 A1 (SONY CORPORATION) 15 March 2018 (2018-03-15) paragraphs [0059], [0090], [0115]	1-13
A	US 2019-0336724 A1 (HUAWEI TECHNOLOGIES CO., LTD.) 07 November 2019 (2019-11-07) claims 1-10; and figures 3-4	1-13
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 30 June 2021		Date of mailing of the international search report 01 July 2021
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon 35208, Republic of Korea Facsimile No. +82-42-481-8578		Authorized officer KIM, Sung Hoon Telephone No. +82-42-481-8710

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/KR2021/000224

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2018-0012330	A1	11 January 2018	US	10726593	B2	28 July 2020
				US	10748313	B2	18 August 2020
				US	2017-0084001	A1	23 March 2017
US	2019-0251721	A1	15 August 2019	US	10748314	B2	18 August 2020
US	2018-0150203	A1	31 May 2018	US	10775962	B2	15 September 2020
				US	2014-0365908	A1	11 December 2014
				US	2015-0268819	A1	24 September 2015
				US	8843853	B1	23 September 2014
				US	9753608	B2	05 September 2017
				US	9880700	B2	30 January 2018
				US	2018-0075490	A1	15 March 2018
				EP	3293691	A1	14 March 2018
				JP	2018-041460	A	15 March 2018
				JP	6547977	B2	24 July 2019
				KR	10-2013493	B1	22 August 2019
				KR	10-2018-0028939	A	19 March 2018
US	2019-0336724	A1	07 November 2019	CN	108604246	A	28 September 2018
				EP	3550450	A1	09 October 2019
				EP	3550450	A4	06 November 2019
				WO	2018-119924	A1	05 July 2018