

METHOD FOR FILTERING IMAGES AND IMAGE HOSTING SERVER

TECHNICAL FIELD

[0001] Various aspects of this disclosure relate to methods for filtering images and image hosting servers.

BACKGROUND

[0002] Via image hosting servers connected to the Internet, users may make images publicly available. However, it is desirable that a server's operator can control which images are published on the server since publishing certain images (herein referred to as prohibited images) may have negative consequences, in particular legal consequences, for the server's operator, for example in case of copyright infringement, but also memory-inefficiency by hosting the same image or multiple versions of the same image (e.g. a cropped version of an image in addition to an image).

[0003] Since each day, a huge number of images may be uploaded, effective automatic approaches for filtering images for an image hosting server are desirable.

SUMMARY

[0004] Various embodiments concern a method for filtering images for an image hosting server including implementing a machine-learning model trained to generate vector representations for input images, generating vector representations of prohibited images by processing each image of a set of prohibited images by the machine-learning model, receiving an image, generating a vector representation for the image by processing the image by the machine-learning model and accepting the image for the image hosting server if the distance of its vector representation to the vector representations of prohibited images is above a predetermined threshold.

[0005] According to one embodiment, the method includes rejecting the image for the image hosting server if the distance of its vector representation to the vector representations of prohibited images is above the predetermined threshold.

[0006] According to one embodiment, the method includes publishing the image by the image hosting server if it has been accepted.

[0007] According to one embodiment, the prohibited images of the set of prohibited images are at least one of images published previously by the image hosting server, copyright-protected images, images not owned by a user from which the image was received and images already hosted by the image hosting server.

[0008] According to one embodiment, the method includes verifying that the image is not prohibited based on metadata of the image and accepting the image for the image hosting only if it is not prohibited.

[0009] According to one embodiment, the method includes verifying that the image is not prohibited by verifying that the image is not watermarked and accepting the image for the image hosting only if it is not prohibited.

[0010] According to one embodiment, the method includes training the machine-learning model to generate vector representations for input images.

[0011] According to one embodiment, the method includes implementing the machine-learning model by base model followed by a head model, wherein the base model is pre-trained and training the machine-learning model includes training the head model from a training data set including training images specific for a domain of images of which the prohibited set of images includes images.

[0012] According to one embodiment, the domain of images is food images or product images.

[0013] According to one embodiment, the base model is trained for images also including images other than the domain of images of which the prohibited set of images includes images.

[0014] According to one embodiment, an image hosting server is provided configured to perform one of the methods described above.

[0015] According to one embodiment, a computer program element is provided including program instructions, which, when executed by one or more processors, cause the one or more processors to perform one of the methods described above.

[0016] According to one embodiment, a computer-readable medium is provided including program instructions, which, when executed by one or more processors, cause the one or more processors to perform one of the methods described above.

[0017] It should be noted that embodiments described in context of one or the methods are analogously valid for the other method and the transport system controller.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

FIG. 1 shows a communication arrangement according to an embodiment.

FIG. 2 shows a flow diagram for filtering images.

FIG. 3 illustrates matching images for an input image.

FIG. 4 shows an example of a machine learning model for generating vector representations from input images.

FIG. 5 shows a flow diagram illustrating a method for filtering images for an image hosting server.

FIG. 6 shows an image hosting server according to an embodiment.

DETAILED DESCRIPTION

[0019] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilized and structural, and logical changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0020] Embodiments described in the context of one of the devices or methods are analogously valid for the other devices or methods. Similarly, embodiments described in the context of a device are analogously valid for a vehicle or a method, and vice-versa.

[0021] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0022] In the context of various embodiments, the articles “a”, “an” and “the” as used with regard to a feature or element include a reference to one or more of the features or elements.

[0023] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0024] In the following, embodiments will be described in detail.

[0025] FIG. 1 shows a communication arrangement 100 according to an embodiment.

[0026] A user 101 using a smartphone 102 may take a photo (i.e. image) 105 of a certain product 103, e.g. a dish in a restaurant, and upload the photo 103 onto a server 104 for publication, e.g. on a site for giving opinions about products, e.g. a site hosting reviews for restaurants. Thus, a user 101 may make images also visible to other consumers (e.g. potential guests of a restaurant), for example to showcase value propositions from the vendor (i.e. here the restaurant owner).

[0027] The user 101 may be a consumer but also a merchant, for example, an owner of a restaurant may himself or herself upload photos 105 about dishes he or she offers to the server 104. The user 101 may also take the photo 105 with a one device and upload it with another, e.g. take a photo with a camera and upload it using a computer.

[0028] It should be noted while the server 104 is described as a single server, its functionality, e.g. image hosting, may in practical application typically be provided by an arrangement of multiple server computers (e.g. implementing a cloud service). Accordingly, the functionality described in the following provided by the server 104 may be understood to be provided by an arrangement of servers or server computers.

[0029] For the operator of the server 104, it is desirable to have mechanisms in place that avoid the upload of photos which may cause problems for the operator, in particular legal problems. One example is that it should be avoided that the server 104 shows photos publicly which are copyright-protected.

[0030] For example, the user 101, being faced with the incentive to provide highly produced quality photos (because they are visible to a large crowd and may be linked to him or her, e.g. via profile) may upload a photo 105 that does not belong to him or her, i.e. may have violated copyright by using a photo 105 that belong to someone else. To avoid that the server's operator likewise violates copyright by making the photo 105 public (e.g. showing it

publicly on a website), according to various embodiments, the server 104 implements a filtering mechanism to filter out such photos.

[0031] It should be noted that avoidance of copyright infringement is only an example. It may also be desirable to avoid double-posting of images on a website because it may be annoying for a user to be shown the same image multiple times when browsing images. It may also be desirable to avoid storage of the same (or very similar) images to save storage space. The server 104 may use the filtering mechanism for this purpose.

[0032] According to various embodiments, a highly accurate, fast, and scalable mechanism for checking whether an image belongs to a prohibited set of images is provided. In particular, when the set of images includes copyright-protected images, the mechanism allows checking the authenticity (in the sense of ownership validation) of an image. For example, using a large archive of known copyright-protected images, allows reliable protection against copyright infringement by the server's operator.

[0033] For example, the server 104 is connected to a data base 106 storing the set of prohibited images (which may be images already published before, copyright-protected images and any images which may be prohibited for any other reason).

[0034] According to various embodiments, the server 104 implements an image filtering process as illustrated in FIG. 2.

[0035] FIG. 2 shows a flow diagram 200 for filtering images.

[0036] The server 104 receives an input image 201, for example corresponding to the photo 105. The input image 201 is for example a newly uploaded photo 105 by a user 101 but the server 104 may also regularly update its content of images and for example read out the image 201 from what it has already stored.

[0037] As explained above, the server 104 is connected to, in this example two, data bases 202, 203 which store a set of prohibited images. This is for example a first data base 202 received (and e.g. published) earlier by the server 104 and a second data base of copyright-protected images, e.g. images published on the Internet by other providers than the server's operator.

[0038] The input image 201 and the images in the data bases 201, 202 may for example be RGB images.

[0039] The server 104 implements a machine-learning model 204 which, for an image supplied to it, generates a vector representation (also referred to as an embedding or a latent representation, in a latent space) for the image.

[0040] For example, the machine-learning model 204 performs a vectorization of each image supplied to it by converting it a one-dimensional array (vector) of a fixed number of floating-point numbers, i.e. a n-dimensional vector with e.g. $n=128$, $n=256$, $n=512$ etc.

[0041] The server 104 processes the images of the data bases 202, 203 by means of the machine learning model 204. The results are vector representations (i.e. embeddings of images in a latent space) which the server 104 stores in an embedding data base 205.

[0042] Regarding the input image 201, the server 104 first performs a pre-screening. This includes checking, in 206, whether metadata (e.g. an EXIF (Exchangeable Image File Format) tag) of the input image 201 indicates that the input image 201 comes from an online source or is subject to copyright. Further, in 207, the server 104 checks whether the input image 201 is watermarked.

[0043] If any one of these checks 206, 207 is positive, the server 104 rejects the input image 201. The checks 206, 207 are to evaluate (and reduce) the risk of copyright infringement. This may include additional checks (and a possible rejection of an image) like for example comparing a location tag of the input image 201 with a location of the user 101 providing the image (and rejecting the image if these two do not correspond) or checking the model of the camera with which the input image was taken 201 (e.g. known from metadata) with the model of the user's smartphone (and rejecting the image if these two do not correspond).

[0044] If the checks 206, 207 are negative, the server 104 supplies the input image 201 to the machine-learning model 204 to get a vector representation (embedding) 208 for the input image.

[0045] In 209, the server 104 performs a vector search, i.e. searches for vector representations in the embedding data base 205 which are close (nearby) to the vector representation 208 for the input image. This means that the server 104 performs a lookup for nearby (e.g. essentially similar) vectors in the embedding data base 205. The vector search is for example performed by a similarity search engine implemented by the server 104.

[0046] If the embedding data base 205 contains a vector which is close to the vector representation 208 for the input image (e.g. its Euclidean distance is below a predefined

threshold), there is high likelihood that the input image 201 is the same as or a variant of the image for which the vector which is close to the vector representation 208 is the vector representation. The server 104 regards such an image of one of the data bases 202, 203 whose vector representation is close to the vector representation of the input image (i.e. distance between the two is below the threshold) as a match (or matching image) for the input image 201.

[0047] By suitable training of the machine-learning model, it can be achieved that after cropping, resizing or rotation, blurring or other convolutions or filtering of an image machine-learning model outputs a vector representation that is very close to vector representation it outputs for the original image. This may be achieved by training the machine-learning model to output the same vector for variants (i.e. versions) of the same picture (e.g. original image, cropped image, rotated image) etc.

[0048] The server 104 may learn the threshold which is used to decide whether two vector representations are close. For example, it may examine the distance of vector representations of variants of the same image (for a large set of training images) and set the threshold (slightly) above the maximum distance observed in this manner.

[0049] From the distance between the vector representation of the input image 201 and the vector representation of a match a match the server 104 may also determine a confidence for the match.

[0050] FIG. 3 illustrates matching images 302, 303, 304 for an input image 301. The first matching image 302 is an exact copy of the input image 301. The distance between the vector representations of the input image 301 and the first matching image 302 is thus zero and the server 104 therefore sets the confidence of matching to 100%.

[0051] The second matching image 303 and the third matching image 304 are versions of the perfect match 302 that have been tampered with (e.g. by cropping and resizing). Their vector representations are still close to the vector representation of the input image (distance below threshold) but not zero. Therefore, the server 104 sets the confidence (depending on the distance) to a value below 100%, e.g. 90%.

[0052] The vector search 209 thus implements an image similarity search and allows the server to identify the top most similar images in the data bases 202, 203, wherein the search results (matches) are insensitive to minor image manipulation by a user (e.g. cropping, blurring, exposure adjustment etc.).

[0053] If the server 104 finds a match for the input image 201, it rejects the image. Otherwise, it accepts the image and, for example, publishes it. By means of the vector search 209, the server 104 thus implements a deep screening of input images.

[0054] FIG. 4 shows an example of a machine learning model 400 for generating vector representations 402, 403 from input images 401.

[0055] The machine learning (ML) model includes a base model 404 and two ML model heads 405, 406. While the base model 404 is trained for a wide range of images the ML model heads 405, 406 are trained for specific domains of images. For example, the first ML model head 405 is trained for fresh food images the second ML model head 406 is trained for images of supermarket products.

[0056] For example, the base model 404 is trained from online images from all domains (e.g. by a number in the range of millions of images). ML model 400 can thus be seen to apply transfer learning from general images to the specific domains of images. The model heads 405, 406 are then trained for the specific domains of images for domain adaptation of the complete ML model 400 to capture finer details that are relevant to images of the specific domains.

[0057] The ML model 400, in particular the base model 404, may be trained in a self-supervised manner, i.e. using unlabelled data. For example, the base model 404 is trained based on that similar images (e.g. versions of the same image) should have similar vector representations and different images should have vector representations as spaced apart as possible.

[0058] In summary, according to various embodiments, a method is provided as illustrated in FIG. 5.

[0059] FIG. 5 shows a flow diagram 500 illustrating a method for filtering images for an image hosting server.

[0060] In 501, a machine-learning model trained to generate vector representations for input images is implemented.

[0061] In 502, vector representations of prohibited images are generated by processing each image of a set of prohibited images by the machine-learning model.

[0062] In 503, an image is received.

[0063] In 504, a vector representation is generated for the image by processing the image by the machine-learning model.

[0064] In 505, the image is accepted for the image hosting server if the distance of its vector representation to the vector representations of prohibited images is above a predetermined threshold.

[0065] The machine learning model is (or includes) for example a neural network. It may for example include a convolutional neural network for image processing, for example followed by one or more fully connected layers.

[0066] The approach of FIG. 5 allows avoiding that an image server accepts (e.g. for storage and publication) images which are the same or similar (i.e. versions of) prohibited images. This allows avoiding copyright infringement and waste of storage (due to multiple storage of essentially the same image). Further, it increases service quality since it is avoided that users are being presented multiple times with essentially the same image when browsing the hosted images.

[0067] The set of prohibited images may in particular be the images that the server already hosts. This allows avoiding that the same image (or variants of the same image) are hosted multiple times.

[0068] The set of prohibited images may at least partially be obtained from users or merchants of a service (e.g. food or product delivery service) to which also the image hosting server belongs.

[0069] The methods described herein may be performed and the various processing or computation units and the devices and computing entities described herein may be implemented by one or more circuits. In an embodiment, a “circuit” may be understood as any kind of a logic implementing entity, which may be hardware, software, firmware, or any combination thereof. Thus, in an embodiment, a “circuit” may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g. a microprocessor. A “circuit” may also be software being implemented or executed by a processor, e.g. any kind of computer program, e.g. a computer program using a virtual machine code. Any other kind of implementation of the respective functions which are described herein may also be understood as a “circuit” in accordance with an alternative embodiment.

[0070] The method of FIG. 6 may for example be performed by an image hosting server 600 as illustrated in FIG. 6.

[0071] FIG. 6 shows an image hosting server 600 according to an embodiment.

[0072] The image hosting server 600, e.g. implemented by a server computer, includes a communication interface 601 (e.g. configured to receive images). The image hosting server 600 further includes a processing unit 602 and a memory 603. The memory 603 may be used by the processing unit 602 to store, for example, the machine learning model. The image hosting server 600 is configured to perform the method of FIG. 5. For example, the memory 603 stores program code which makes the image hosting server 600 perform the method of FIG. 6.

[0073] While the disclosure has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A method for filtering images for an image hosting server comprising:
implementing a machine-learning model trained to generate vector representations for input images;
generating vector representations of prohibited images by processing each image of a set of prohibited images by the machine-learning model;
receiving an image;
generating a vector representation for the image by processing the image by the machine-learning model; and
accepting the image for the image hosting server if the distance of its vector representation to the vector representations of prohibited images is above a predetermined threshold.
2. The method of claim 1, comprising rejecting the image for the image hosting server if the distance of its vector representation to the vector representations of prohibited images is below the predetermined threshold.
3. The method of claim 1 or 2, comprising publishing the image by the image hosting server if it has been accepted.
4. The method of any one of claims 1 to 3, wherein the prohibited images of the set of prohibited images are at least one of images published previously by the image hosting server, copyright-protected images, images not owned by a user from which the image was received and images already hosted by the image hosting server.
5. The method of any one of claims 1 to 4, comprising verifying that the image is not prohibited based on metadata of the image and accepting the image for the image hosting only if it is not prohibited.

6. The method of any one of claims 1 to 5, comprising verifying that the image is not prohibited by verifying that the image is not watermarked and accepting the image for the image hosting only if it is not prohibited.
7. The method of any one of claims 1 to 6, comprising training the machine-learning model to generate vector representations for input images.
8. The method of claim 7, comprising implementing the machine-learning model by base model followed by a head model, wherein the base model is pre-trained and training the machine-learning model comprises training the head model from a training data set comprising training images specific for a domain of images of which the prohibited set of images comprises images.
9. The method of claim 8, wherein the domain of images is food images or product images.
10. The method of claim 8, wherein the base model is trained for images also including images other than the domain of images of which the prohibited set of images comprises images.
11. An image hosting server configured to perform the method of any one of claims 1 to 10.
12. A computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 1 to 10.
13. A computer-readable medium comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 1 to 10.