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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 December 2021 (30.12.2021)

WIPO | PCT

(10) International Publication Number
WO 2021/263019 A1

(51) International Patent Classification:
G06F 16/583 (2019.01) *G06K 9/00* (2006.01)

(21) International Application Number:
PCT/US2021/038949

(22) International Filing Date:
24 June 2021 (24.06.2021)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
16/912,366 25 June 2020 (25.06.2020) US

(71) Applicant: **AMAZON TECHNOLOGIES, INC.** [US/US]; 410 Terry Avenue North, Seattle, Washington 98109-5210 (US).

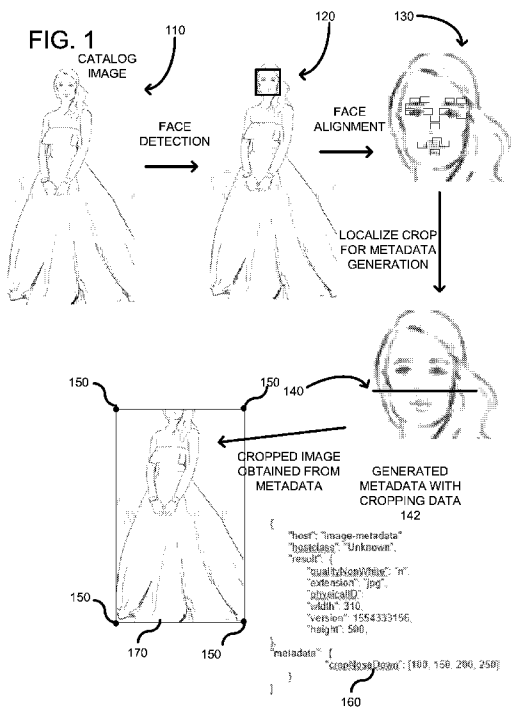
(72) Inventors: **STERNIG, Sabine**; c/o Amazon Technologies, Inc., 410 Terry Avenue North, Seattle, Washington 98109-5210 (US). **STAUDIGL, Luitpold**; c/o Amazon Technologies, Inc., 410 Terry Avenue North, Seattle, Washington 98109-5210 (US). **BARBACHANO, Pablo**; c/o Amazon Technologies, Inc., 410 Terry Avenue North, Seattle, Washington 98109-5210 (US). **WALLIS, Thomas Sydney Austin**; c/o Amazon Technologies, Inc., 410 Terry Avenue North, Seattle, Washington 98109-5210 (US). **JAVED, Muhammad Bilal**; c/o Amazon Technologies, Inc., 410 Terry Avenue North, Seattle, Washington 98109-5210 (US). **DONOSER, Michael**; c/o Amazon Technologies, Inc., 410 Terry Avenue North, Seattle, Washington 98109-5210 (US).

(74) Agent: **SCOTTI, Robert F.**; One World Trade Center Ste. 1600, 121 SW Salmon St., Portland, Oregon 97204 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, 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,

(54) Title: IMAGE CROPPING USING PRE-GENERATED METADATA



(57) Abstract: In operating an ecommerce marketplace, pre-generated metadata is used to transform images after a request is received, in real-time, and with low latency. To accomplish the high-speed image transformation, an offline annotation of relevant image features is obtained using machine learning and metadata is stored based upon the results. The metadata is then used to perform the high-speed image transformation at request time. The transformations can include image cropping, adjustment in saturation, contrast, brightness, extracting portions of the image, etc. In one example, the metadata includes a bounding box giving coordinates of how to crop an image so that the image can be cropped without analyzing content or context of the image. Instead, to obtain the coordinates of the bounding box, content and context of the image are analyzed in pre-request processing. In this way, when the request is received, the more difficult and time-consuming processing is already completed.

WO 2021/263019 A1

[Continued on next page]

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

IMAGE CROPPING USING PRE-GENERATED METADATA

BACKGROUND

[0001] In an ecommerce marketplace, customers view millions of images per day, such as images of fashion. Third-party vendors supply images of their products to the ecommerce marketplace. As customer preferences and trends vary over time, website developers want to quickly test new ideas on website retail pages. While it is usually straightforward to implement low-level image transformations, such as resizing or changing the image quality, content aware image processing creates novel challenges. Replacing images in an image catalogue typically involves coordination with a large number of the third-party vendors and significant quality assurance efforts. Fast iterations of image replacement are difficult to implement. Performing image alterations at request time allows more control over the process, but the latency concerns and the number of image alterations makes request-time alterations impractical.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] FIG. 1 shows an example of generation of metadata used for request-time image alterations.

[0003] FIG. 2 is a system according to one embodiment for generating metadata used for request-time image alterations, wherein a request is shown having a cropping style tag.

[0004] FIG. 3 is a system according to another embodiment, wherein metadata is generated for a catalog of images.

[0005] FIG. 4 is a system according to another embodiment, wherein a metadata generator receives a notification to process an image and generate metadata used for cropping the image.

[0006] FIG. 5 is a flowchart according to one embodiment for image cropping using stored metadata.

[0007] FIG. 6 is a flowchart according to another embodiment for generating metadata used for cropping an image.

[0008] FIG. 7 is a flowchart according to yet another embodiment for using a cropping style tag in a request.

[0009] FIG. 8 depicts a generalized example of a suitable computing environment in which the described innovations may be implemented.

DETAILED DESCRIPTION

[0010] In operating an ecommerce marketplace, pre-generated metadata is used to crop images after a request is received, in real-time, and with low latency. The meta data is generated through a Machine Learning model can be used for image transformations such as: adjustment of pixels based on image statistics (e.g., saturation, brightness, contrast) or applying kernel operations (e.g. sharpening) either for the image as a whole or for parts of the image. Accordingly, the image transformations happen much quicker than the generation of the required meta information that is used for the transformation. A specific example could be: whenever a women in a red dress is found in an image, the saturation of the image should be increased to increase the intensity of the red dress. If a woman in a green dress is found, then the saturation of the image should be decreased and images should appear "flatter" than they were originally. Adjusting the saturation of the image is dependent on running a specific computer vision task that requires semantic understanding of the image. Thus, embodiments herein can perform low-level computer vision tasks not requiring any understanding of the image based on the extraction of a higher level semantic image understanding.

[0011] In one example, the metadata includes a bounding box giving coordinates of how to crop an image so that the image can be cropped without analyzing content or context of the image. Instead, to obtain the coordinates of the bounding box, content and context of the image are analyzed in pre-request processing. In this way, when the request is received, the more difficult and time-consuming processing is already completed. The system achieves the required accuracy through available face-detection models and heuristics. When a user requests a webpage, the original images (uncropped) can by dynamically cropped based upon the request and at high speed. To accomplish the high-speed cropping, an offline annotation of relevant image features is obtained using machine learning and metadata is stored based upon the results. The metadata is then used to perform the high-speed cropping at run time.

[0012] In one example, for cropping below the nose, machine learning is used to identify a consistent location. Generally, face detection is used to locate a face in an image and facial landmark detection is used to locate features of the located face. Face detection has been used in Human-Computer Interaction (HCI), video-surveillance, and security systems. Feature-based approaches have also been developed using boosted classifiers to identify features that describe faces. Two-stage methods, such as Faster R-CNN, and single-stage methods, such as RetinaNet, can be used. Facial landmark detection, which is also known as facial keypoint detection or face alignment, is the task of identifying predefined facial landmarks, such as the corners of eyes, eyebrows or the tip of the nose.

[0013] Facial detection for images used in a retail online marketplace are adapted to detect faces in catalog images. A Single-Shot Scale-Invariant Face Detector (S3FD) can be used that applies extra convolutional layers and detection layers, wherein each detection layer is associated with a specific scale anchor to detect corresponding scale faces. A stride size of anchor-associated layers ensures that different scales of faces have adequate features for detection at corresponding anchor-associated layers and that different scales of anchors have a same density on the image. Another facial detection algorithm is RetinaFace wherein additional segmentation information in a segmentation branch improves object detection results. RetinaFace employs supervised and self-supervised multi-task learning in parallel with the existing box classification and regression branches. Each positive anchor predicts a face score, a face box, five facial landmarks, and dense 3D position and correspondence of each facial pixel. A variety of facial landmark localization algorithms can be used. In one, the framework is based on gradient boosting for learning the ensemble of regression trees where a sparse subset of pixel intensities is used to regress the location of facial landmarks.

[0014] FIG. 1 is an example embodiment wherein metadata is generated using content and context analysis. As explained further below, the metadata is "pre-generated" relative to a request. Thus, the metadata is generated before a request and later used after a request is received to perform fast cropping of the image without the need of further content and context analysis. A catalog image 110 is supplied from a third-party seller for an e-commerce marketplace. The image 110 is of a person wearing clothing for sale. At 120, facial detection is performed using any of the available

methods described above. Likewise, at 130, facial landmark localization algorithms can be used to find facial alignment and facial features, such as corners of the eyes, the nose, and the mouth. At 140, a below-the-nose line is generated at the base of the person's nose. The below-the-nose line 140 is then used to generate metadata 142 including a bounding box defined by points 150. The metadata 142 includes the bounding box coordinates at 160, which are 4 pixel coordinates associated with the image. As described further below, future requests that ask for the image 110 and with below-the-nose cropping, can skip the face detection 120, face alignment 130, and the below-the-nose line 140, and merely use the pre-generated metadata 160 to quickly generate the cropped image 170. In any of the above processes, machine learning can be used to facilitate identification of facial features and to generate the metadata.

[0015] FIG. 2 is an example system 200 for providing cropped images with low latency. A client browser 210 executing on a client computer accesses an ecommerce marketplace backend 220 to obtain images to display to a user. The images can include multiple images a person, such as was shown in FIG. 1, wherein the person is modeling clothing, shoes, etc. The ecommerce marketplace backend 220 includes a content delivery cache 230, a metadata generator and image cropper 232, a metadata database 234, and an image database 236. The ecommerce marketplace backend 220 can be modified in multiple ways, such as by combining databases 234, 236, eliminating cache 230, splitting the metadata generator from the image cropping 232, etc. Any of the described blocks includes software executing on one or more server computers, such as a computer of FIG. 8 described below. The client browser 210 transmits a request 240 that is received by the content delivery cache 230 (after passing through an endpoint, authentication, etc. (not shown)). The request 240 is, for example, an HTML request for an image and includes an image identifier and a cropping style tag. The cropping style tag requests that the image be modified by cropping the image using a variety of cropping techniques, such as cropping from the nose down. Cropping an image, in this context, refers to removal of a section (e.g., entire sequential rows) of an image to modify the image appearance, as opposed to merely removing selected pixels, which merely sharpens the image appearance. The content delivery cache 230 performs a lookup on the image identifier to determine if the image is available in the cache 230. If not, a cache miss notification 250 is transmitted to the metadata generator and image cropping 232. The metadata generator and image cropping 232 requests the image at 252 from the image database 236 using the image

identifier and requests metadata from the database 234 using the image identifier 254. These requests 252, 254 can occur in parallel when there are separate databases 234, 236, as shown. The image is returned 256 in response to the request 252 and the metadata 260 is returned that includes the results of predetermined cropping analysis. For example, the metadata can be bounding box coordinates obtained from previous analysis performed on the image including face detection, face alignment, nose identification, etc. Using the metadata 260, the metadata generator and image cropping 232 quickly crops the image with low latency and in accordance with the style tag of the request 240. The cropped image is returned at 270 to the content delivery cache 230, which caches the cropped image for future requests. The cropped image is then returned at 280 to the client browser 210 for display to the user.

[0016] Although FIG. 2 is described in relation to cropping, a variety of image transformations can be performed in addition to or alternatively to cropping. For example, the image transformations can include changes to brightness, saturation, color space, resizing, segmenting, extracting portions of image content, etc. Additionally, transformation rules can be derived from the metadata and the rules used in the image transformation. The metadata generator and image cropping 232 can include logic following a set of rules or policies on whether to use the metadata. In one example, the logic decides whether the image transformation takes longer than a predetermined period of time, and, if so, then the metadata is used. Otherwise, the transformation occurs without retrieving the metadata. The metadata can further be used for decisions whether to delete the image, flag the image (e.g., as inappropriate) or use the meta data in processing steps that are not directly linked to image transformations, but could also be "non-image transformation-related computer processing" (e.g., enrich the search catalog).

[0017] As the generation of the meta data based on semantic image understanding is computationally complex and expensive, heuristics or rule-based mechanisms can be put in place to ensure that these operations are only performed when necessary. To this end, information that is available for the image set (e.g. a product category, a human classification or the like) is considered and pre-processed before the computer vision models are applied. For example, to increase the saturation of red dresses whenever they are worn by a woman, a classifier/model need not be run on all image material, but only on the category "dresses" as only this category is relevant for the task.

Extraction of features can be performed through computer vision for images that, based on additional classification and data that is available, update a list as images are added or removed, which reduces the computational complexity of processing the images.

[0018] Any of the described requests, such as requests 240, 252, 254, etc., can be implemented using an API. The described blocks, such as the metadata generator 232, can be a service in a cloud computing environment. A service is a software function provided at a network address over the web or the cloud. Clients initiate service requests to servers and servers process the requests and return appropriate responses. The client service requests are typically initiated using, for example, an API request. For purposes of simplicity, service requests will be generally described below as API requests, but it is understood that other service requests can be made. An API request is a programmatic interface to a defined request-response message system, typically expressed in JSON or XML, which is exposed via the web—most commonly by means of an HTTP-based web server. Thus, in certain implementations, an API can be defined as a set of Hypertext Transfer Protocol (HTTP) request messages, along with a definition of the structure of response messages, which can be in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format. The API can specify a set of functions or routines that perform an action, which includes accomplishing a specific task or allowing interaction with a software component. When a service receives the API request from a client device, the service can generate a response to the request and send the response to the endpoint identified in the request.

[0019] FIG. 3 illustrates a system 300 according to another embodiment, wherein new images are detected and metadata is generated for future cropping of the new images. A catalog 310 includes available images in the ecommerce marketplace and typically contains millions of images to be viewed. Different style tags are received to crop the images. The catalog 310 is a database or a server computer having storage thereon. A filter 320 is used to determine categories of images that are typically cropped in websites. For example, the filter 320 can filter fashion-based photos, such as models wearing women's dresses, shoes, etc. Alternatively, the catalog 310 includes identifiers of images, rather than the images themselves. The filter 320 is a server computer executing software to perform the filter function. A metadata generator 330 detects the new image in the catalog 310 via a notification from the filter 320. The metadata generator 330 uses the received

identifier to request an image at 332. The image is then returned at 334. The metadata generator 330 performs one or more image cropping based on style tags supported. The image cropping requires image content and context analysis, which is compute intensive. Such backend processing occurs prior to browser requests such that the browser requests, when made, require substantially less compute time. The metadata generated by the metadata generator 330 is stored together with the image identifier in the metadata database 350. Example of metadata is a bounding box coordinates, line coordinates, predetermined pixel coordinates, etc.

[0020] FIG. 4 is an example of another embodiment of a system 400 wherein metadata is calculated for new images and stored for later use. At 410, a catalog has a plurality of original, uncropped images. The images are transmitted to an image upload pipeline 420, which detects new images not previously analyzed and transmits a notification to a metadata generator 430. The notification includes an identifier of the image, a category, a browse mode, etc. The metadata generator 430 obtains the image from the pipeline 420 or can alternatively obtain the image from a separate database, such as described above in FIG. 2. The metadata generator 430 generates the metadata and stores the metadata in association with the image identifier in the metadata database 450. The metadata is thereby available for later use in response to requests including a cropping stylus tag. A single image can have multiple different metadata coordinates, each associated with a different style tag. Thus, requests can include any supported style tags and the image is cropped using the retrieved metadata and only the metadata associated with the style tag of the request is used in the cropping, while metadata for other style tags is ignored. For example, a single image can have metadata associated with multiple bounding boxes, one for below the nose and another for above the neck. According to the style tag of the request, only the appropriate metadata is chosen

[0021] FIG. 5 is a flowchart according to one embodiment for generating a transformed image using stored metadata. In process block 510, a request is received for an image including an image identifier and a style tag. The image identifier is any desired identifier, such as a Globally Unique Identifier (GUID), and the style tag is a markup language tag, such as an HTML style tag, that indicates to the browser to transform an image. The style tag can be a variety of potential cropping types, such as a nose-down crop, mouth-up crop, etc. In one example, a facial feature is chosen and cropping is based on identification of the facial feature in the image. The cropping typically starts

with a top or bottom of the identified facial feature. Other transformation of the image can also be associatied with the style tag, such as modifying saturation, modifying brightness, modifying contrast, changing to black and white, etc. At process block 520, an image is retrieved that is associated with the image identifier. For example, in FIG. 2, the metadata generator 232 retrieves an image from the image database 236. In process block 530, metadata is retrieved using the image identifier. Thus, image analysis previously performed on the image is retrieved from a metadata database, such as the metadata database 234 (FIG. 2). In process block 540, the retrieved image is transformed in accordance with the metadata. Thus, the metadata provides specific information on how to transform the image to conform with the style tag. For example, in FIG. 1, the metadata 160 provides a bounding box that starts with the nose-down cropping to result in the image 170. Cropping using a bounding-box result is performed quickly and no facial detection or face alignment is needed after the request is made, as the metadata was pre-generated (generated prior to the request or offline). At process block 550, the transformed image is transmitted in response to the request. For example, in FIG. 2, the image cropping occurs by server computer 232, which crops the image in conformance with the metadata 260 and transmits the response to the content delivery cache 230, which caches the resultant image, and then transmits the image 280 as a response. Thus, the browser 210 receives an image that is cropped at request time in conformance with the cropping style tag.

[0022] FIG. 6 is a flowchart according to one embodiment for generating metadata used for future transformation (e.g., cropping, adjustment in saturation, contrast, brightness, extracting portions of the image, etc.) of an image. In process block 610, an original image is received. Typically, the original image is stored unmodified and the transformation occurs dynamically at run-time. In process block 620, image analysis is performed on the image associated with transforming of the image. Such a transformation requires content and context analysis and recognition. In one example, facial feature recognition or identification of body parts is performed. In process block 630, metadata is generated resulting from the image analysis. Thus, a result of the image analysis is summarized as metadata, such as a bounding box, a line indicating a location of a nose or other facial feature, etc. In process block 640, the metadata is stored for the image and is retrieved later when a request is received to transform the image. For example, the metadata is stored in the

metadata database 234 (FIG. 2). In process block 650, upon receiving a request for the image, the image is transformed using the stored metadata.

[0023] FIG. 7 is a flowchart according to another embodiment for dynamically transforming an image in response to a request including a style tag. In process block 710, a request is received for an image of a person. The request includes an image identifier and a style tag. The style tag provides an indication of how to dynamically transform the image. In process block 720, a transformed version of the image is transmitted in accordance with the style tag. Thus, the style tag is used to identify a portion of a person in the image that is to be shown as a result of the transformation. Another portion of the person is removed from the image. The transformation of the image is performed after the request using stored metadata that is used to understand the content of the image and how to modify the image in relation to that content.

[0024] FIG. 8 depicts a generalized example of a suitable computing environment 800 in which the described innovations may be implemented. The computing environment 800 is not intended to suggest any limitation as to scope of use or functionality, as the innovations may be implemented in diverse general-purpose or special-purpose computing systems. For example, the computing environment 800 can be any of a variety of computing devices (e.g., desktop computer, laptop computer, server computer, tablet computer, etc.).

[0025] With reference to FIG. 8, the computing environment 800 includes one or more processing units 810, 815 and memory 820, 825. In FIG. 8, this basic configuration 830 is included within a dashed line. The processing units 810, 815 execute computer-executable instructions. A processing unit can be a general-purpose central processing unit (CPU), processor in an application-specific integrated circuit (ASIC) or any other type of processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power. For example, FIG. 8 shows a central processing unit 810 as well as a graphics processing unit or co-processing unit 815. The tangible memory 820, 825 may be volatile memory (e.g., registers, cache, RAM), non-volatile memory (e.g., ROM, EEPROM, flash memory, etc.), or some combination of the two, accessible by the processing unit(s). The memory 820, 825 stores software 880 implementing one or more innovations described herein, in the form of computer-executable

instructions suitable for execution by the processing unit(s). The computing environment 800 can be used, for example, to implement the metadata generator and image cropping 232.

[0026] A computing system may have additional features. For example, the computing environment 800 includes storage 840, one or more input devices 850, one or more output devices 860, and one or more communication connections 870. An interconnection mechanism (not shown) such as a bus, controller, or network interconnects the components of the computing environment 800. Typically, operating system software (not shown) provides an operating environment for other software executing in the computing environment 800, and coordinates activities of the components of the computing environment 800.

[0027] The tangible storage 840 may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, DVDs, or any other medium which can be used to store information in a non-transitory way and which can be accessed within the computing environment 800. The storage 840 stores instructions for the software 880 implementing one or more innovations described herein.

[0028] The input device(s) 850 may be a touch input device such as a keyboard, mouse, pen, or trackball, a voice input device, a scanning device, or another device that provides input to the computing environment 800. The output device(s) 860 may be a display, printer, speaker, CD-writer, or another device that provides output from the computing environment 800.

[0029] The communication connection(s) 870 enable communication over a communication medium to another computing entity. The communication medium conveys information such as computer-executable instructions, audio or video input or output, or other data in a modulated data signal. A modulated data signal is a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media can use an electrical, optical, RF, or other carrier.

[0030] Although the operations of some of the disclosed methods are described in a particular, sequential order for convenient presentation, it should be understood that this manner of description encompasses rearrangement, unless a particular ordering is required by specific language set forth

below. For example, operations described sequentially may in some cases be rearranged or performed concurrently. Moreover, for the sake of simplicity, the attached figures may not show the various ways in which the disclosed methods can be used in conjunction with other methods.

[0031] Any of the disclosed methods can be implemented as computer-executable instructions stored on one or more computer-readable storage media (e.g., one or more optical media discs, volatile memory components (such as DRAM or SRAM), or non-volatile memory components (such as flash memory or hard drives)) and executed on a computer (e.g., any commercially available computer, including smart phones or other mobile devices that include computing hardware). The term computer-readable storage media does not include communication connections, such as signals and carrier waves. Any of the computer-executable instructions for implementing the disclosed techniques as well as any data created and used during implementation of the disclosed embodiments can be stored on one or more computer-readable storage media. The computer-executable instructions can be part of, for example, a dedicated software application or a software application that is accessed or downloaded via a web browser or other software application (such as a remote computing application). Such software can be executed, for example, on a single local computer (e.g., any suitable commercially available computer) or in a network environment (e.g., via the Internet, a wide-area network, a local-area network, a client-server network (such as a cloud computing network), or other such network) using one or more network computers.

[0032] For clarity, only certain selected aspects of the software-based implementations are described. Other details that are well known in the art are omitted. For example, it should be understood that the disclosed technology is not limited to any specific computer language or program. For instance, aspects of the disclosed technology can be implemented by software written in C++, Java, Perl, any other suitable programming language. Likewise, the disclosed technology is not limited to any particular computer or type of hardware. Certain details of suitable computers and hardware are well known and need not be set forth in detail in this disclosure.

[0033] It should also be well understood that any functionality described herein can be performed, at least in part, by one or more hardware logic components, instead of software. For example, and without limitation, illustrative types of hardware logic components that can be used include Field-programmable Gate Arrays (FPGAs), Program-specific Integrated Circuits (ASICs), Program-

specific Standard Products (ASSPs), System-on-a-chip systems (SOCs), Complex Programmable Logic Devices (CPLDs), etc.

[0034] Furthermore, any of the software-based embodiments (comprising, for example, computer-executable instructions for causing a computer to perform any of the disclosed methods) can be uploaded, downloaded, or remotely accessed through a suitable communication means. Such suitable communication means include, for example, the Internet, the World Wide Web, an intranet, software applications, cable (including fiber optic cable), magnetic communications, electromagnetic communications (including RF, microwave, and infrared communications), electronic communications, or other such communication means.

[0035] The following numbered paragraphs provide additional features and advantages of the embodiments described herein:

[0036] 1. A method of cropping an image using stored metadata, the method comprising: receiving a request for an image, the request including an image identifier and a style tag indicating how to transform the image;

retrieving the image associated with the image identifier;

retrieving metadata using the image identifier, the metadata indicating how to transform the image in accordance with the style tag;

transforming the image using the metadata; and

transmitting the transformed image in response to the request.

[0037] 2. The method of paragraph 1, wherein the metadata includes transformation rules for modifying the image.

[0038] 3. The method of paragraph 2, wherein the transformation rules including rules for cropping the image using predetermined bounding box coordinates or pixel coordinates stored for use in cropping the image.

[0039] 4. The method of paragraph 1, wherein the metadata is generated prior to receiving the request by performing machine learning on the image and storing pixel coordinates as the metadata.

[0040] 5. The method of paragraph 1, wherein the style tag is associated with a nose-down cropping and the metadata includes pixel coordinates for performing cropping of the image below a nose of a person in the image.

[0041] 6. A method, comprising:

receiving an image;

performing image analysis on the image associated with cropping of the image;

generating metadata resulting from the image analysis;

storing the metadata for the image to be used when receiving a request for the cropping of the image;

upon receiving a request for the image, transforming the image using the stored metadata.

[0042] 7. The method of paragraph 6, wherein the metadata is stored in association with a cropping style tag and further including retrieving the metadata in response to a request including the cropping style tag.

[0043] 8. The method of paragraph 7, further including receiving a request for an image, wherein the request includes the cropping style tag, and performing the cropping of the image in response to the request by retrieving the stored metadata associated with the image and cropping the image in conformance with the metadata.

[0044] 9. The method of paragraph 8, further including filtering images from a catalog and generating the metadata for the filtered images.

[0045] 10. The method of paragraph 6, wherein the metadata is stored in association with an identifier of the image and the image is stored separately from the metadata using the identifier.

[0046] 11. The method of paragraph 6, wherein the generated metadata is a bounding box used for the cropping of the image.

[0047] 12. The method of paragraph 6, wherein the generated metadata includes predetermined pixel coordinates associated with the image.

[0048] 13. The method of paragraph 6, further including receiving a request for an image, wherein the request includes a style tag that is associated with a nose-down cropping and retrieving the metadata and using the metadata to perform the nose-down cropping on the image.

[0049] 14. A method, comprising:

receiving a request for an image of a person, wherein the request includes a cropping style tag; and

transmitting a cropped version of the image which has only a portion of the person in the image in accordance with the cropping style tag.

[0050] 15. The method of paragraph 14, wherein the image is an original image and the transmitting of the cropped version of the image has a portion of the original image removed.

[0051] 16. The method of paragraph 14, wherein the cropping style tag is a nose-down cropping and the portion of the person in the cropped version of the image is from a nose of the person down to feet of the person.

[0052] 17. The method of paragraph 14, further including performing machine learning analysis on the image to understand content and context of the person within the image including facial features and generating metadata usable to perform cropping of the image to generate the cropped version of the image.

[0053] 18. The method of paragraph 17, wherein the metadata is prestored for use when receiving the request.

[0054] 19. The method of paragraph 17, wherein the metadata is bounding box coordinates used for cropping of the image.

[0055] 20. The method of paragraph 14, wherein the request is from a browser of a client computer.

[0056] The disclosed methods, apparatus, and systems should not be construed as limiting in any way. Instead, the present disclosure is directed toward all novel and nonobvious features and aspects of the various disclosed embodiments, alone and in various combinations and subcombinations with one another. The disclosed methods, apparatus, and systems are not limited to any specific aspect or feature or combination thereof, nor do the disclosed embodiments require that any one or more specific advantages be present or problems be solved.

[0057] In view of the many possible embodiments to which the principles of the disclosed invention may be applied, it should be recognized that the illustrated embodiments are only examples of the invention and should not be taken as limiting the scope of the invention. We therefore claim as our invention all that comes within the scope of these claims.

CLAIMS

WHAT IS CLAIMED IS:

1. A method, comprising:

receiving an image;

performing image analysis on the image associated with cropping of the image;

generating metadata resulting from the image analysis;

storing the metadata for the image to be used when receiving a request for the cropping of the image;

upon receiving a request for the image, transforming the image using the stored metadata.

2. The method according to claim 1, wherein the metadata is stored in association with a cropping style tag and further including retrieving the metadata in response to a request including the cropping style tag.

3. The method according to claims 1-2, further including receiving a request for an image, wherein the request includes the cropping style tag, and performing the cropping of the image in response to the request by retrieving the stored metadata associated with the image and cropping the image in conformance with the metadata.

4. The method according to any preceding claims, further including filtering images from a catalog and generating the metadata for the filtered images.

5. The method according to any preceding claims, wherein the metadata is stored in association with an identifier of the image and the image is stored separately from the metadata using the identifier.

6. The method according to any preceding claims, wherein the generated metadata is a bounding box used for the cropping of the image.

7. The method according to any preceding claims, wherein the generated metadata includes predetermined pixel coordinates associated with the image.

8. The method according to any preceding claims, further including receiving a request for an image, wherein the request includes a style tag that is associated with a nose-down cropping and retrieving the metadata and using the metadata to perform the nose-down cropping on the image.

9. One or more non-transitory computer-readable storage media storing computer-executable instructions for causing a computing system to perform a method, the method comprising:

receiving a request for an image of a person, wherein the request includes a cropping style tag; and

transmitting a cropped version of the image which has only a portion of the person in the image in accordance with the cropping style tag.

10. The one or more non-transitory computer-readable storage media of claim 9, wherein the image is an original image and further including executable instructions for transmitting the cropped version of the image with a portion of the original image removed.

11. The one or more non-transitory computer-readable storage media according to claims 9-10, further including executable instructions for cropping from a nose-down including cropping of the person in the cropped version of the image from a nose of the person down to feet of the person.

12. The one or more non-transitory computer-readable storage media according to claims 9-11, further including executable instructions to perform machine learning analysis on the image to understand content and context of the person within the image including facial features and to generate metadata usable to perform cropping of the image to generate the cropped version of the image.

13. The one or more non-transitory computer-readable storage media according to claim 12, wherein the metadata is prestored for use when receiving the request.

14. The one or more non-transitory computer-readable storage media according to claims 12-13, further including executable instructions for storing bounding box coordinates in metadata used for cropping of the image.

15. The one or more non-transitory computer-readable storage media according to claims 9-14, wherein the request is from a browser of a client computer.

FIG. 1

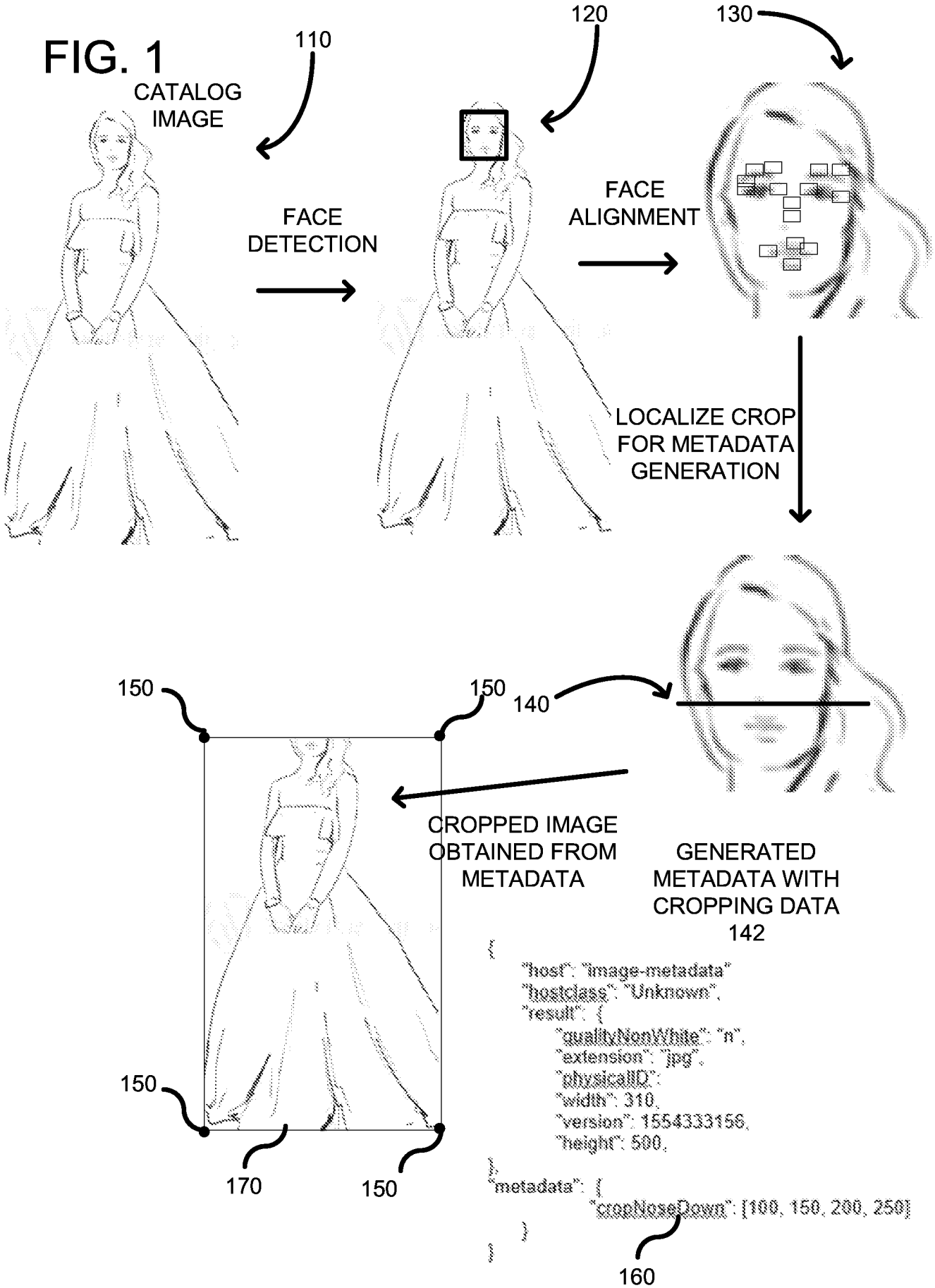
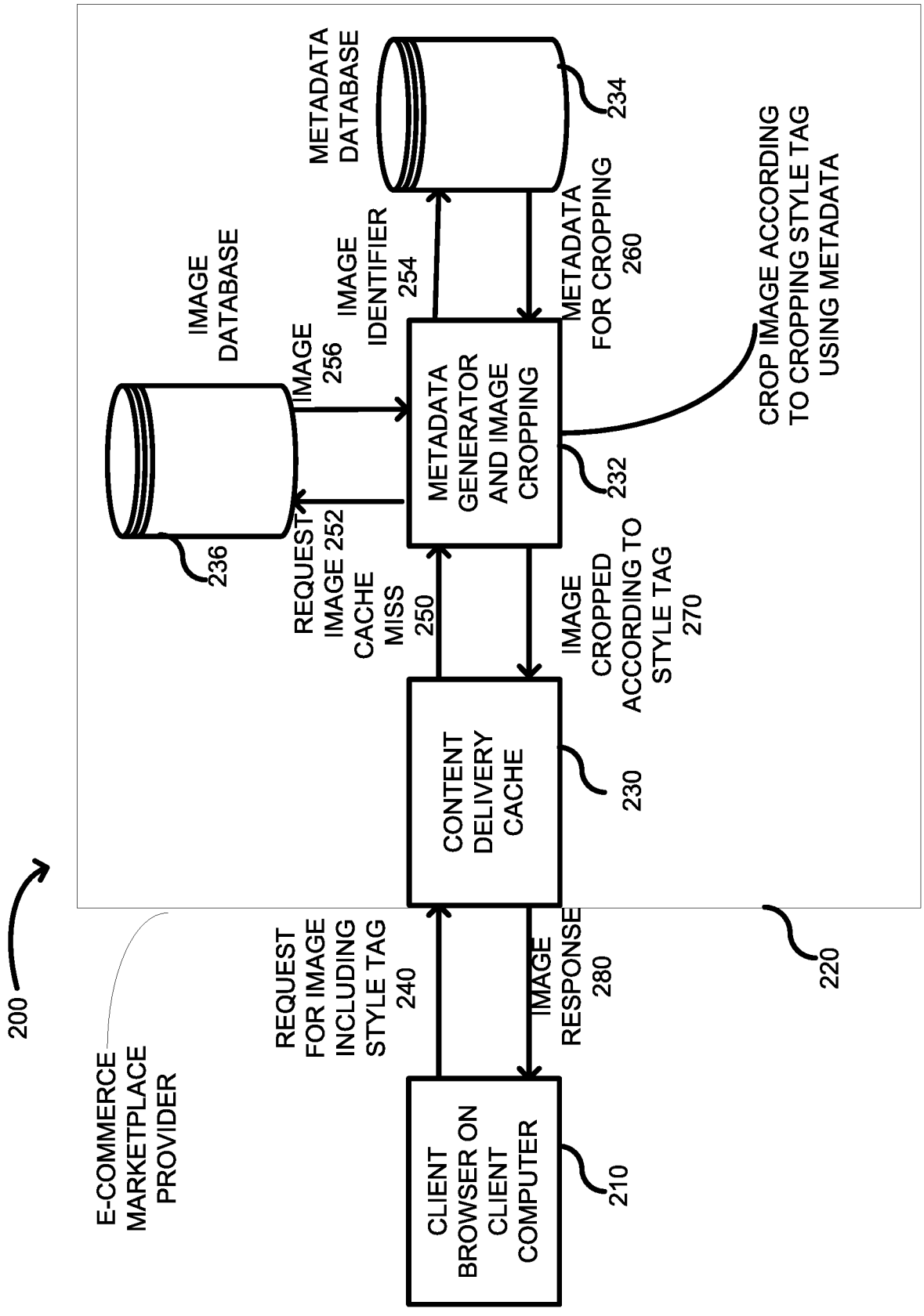

110
CATALOG IMAGE
120
130
FACE DETECTION
FACE ALIGNMENT
LOCALIZE CROP FOR METADATA GENERATION
140
150
150
150
150
170
160
CROPPED IMAGE OBTAINED FROM METADATA
GENERATED METADATA WITH CROPPING DATA 142
{
"host": "image-metadata"
"hostclass": "Unknown",
"result": {
"qualityNonWhite": "n",
"extension": "jpg",
"physicalID":
"width": 310,
"version": 1554333156,
"height": 500,
},
"metadata": {
"cropNoseDown": [100, 150, 200, 250]
}
}

FIG. 2


200
E-COMMERCE MARKETPLACE PROVIDER
CLIENT BROWSER ON CLIENT COMPUTER
210
REQUEST FOR IMAGE INCLUDING STYLE TAG 240
IMAGE RESPONSE 280
CONTENT DELIVERY CACHE
230
REQUEST IMAGE 252 CACHE MISS 250
IMAGE CROPPED ACCORDING TO STYLE TAG 270
METADATA GENERATOR AND IMAGE CROPPING
232
IMAGE IDENTIFIER 254
IMAGE 256
METADATA FOR CROPPING 260
CROP IMAGE ACCORDING TO CROPPING STYLE TAG USING METADATA
IMAGE DATABASE
236
METADATA DATABASE
234
220

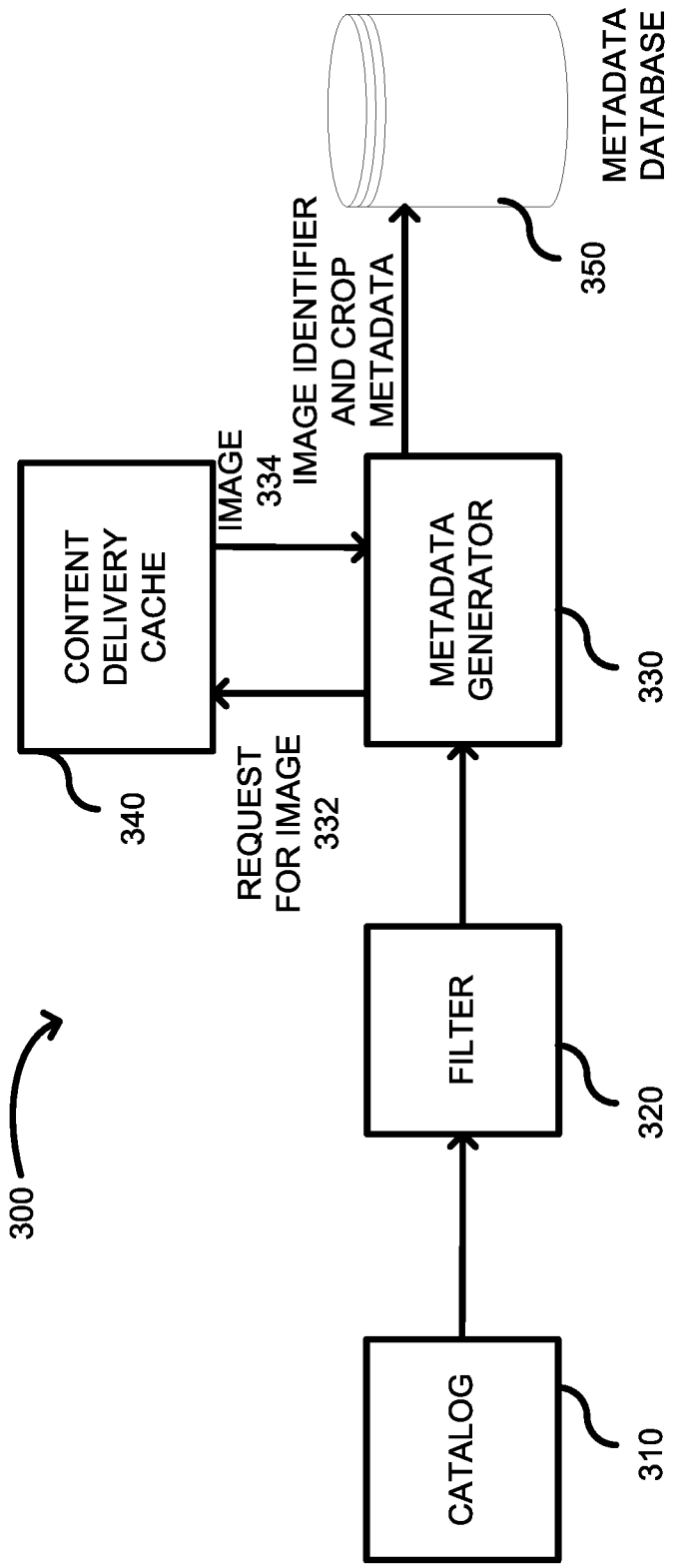

300
CATALOG
310
FILTER
320
METADATA GENERATOR
330
CONTENT DELIVERY CACHE
340
METADATA DATABASE
350
REQUEST FOR IMAGE
332
IMAGE
334
IMAGE IDENTIFIER AND CROP METADATA

FIG. 3

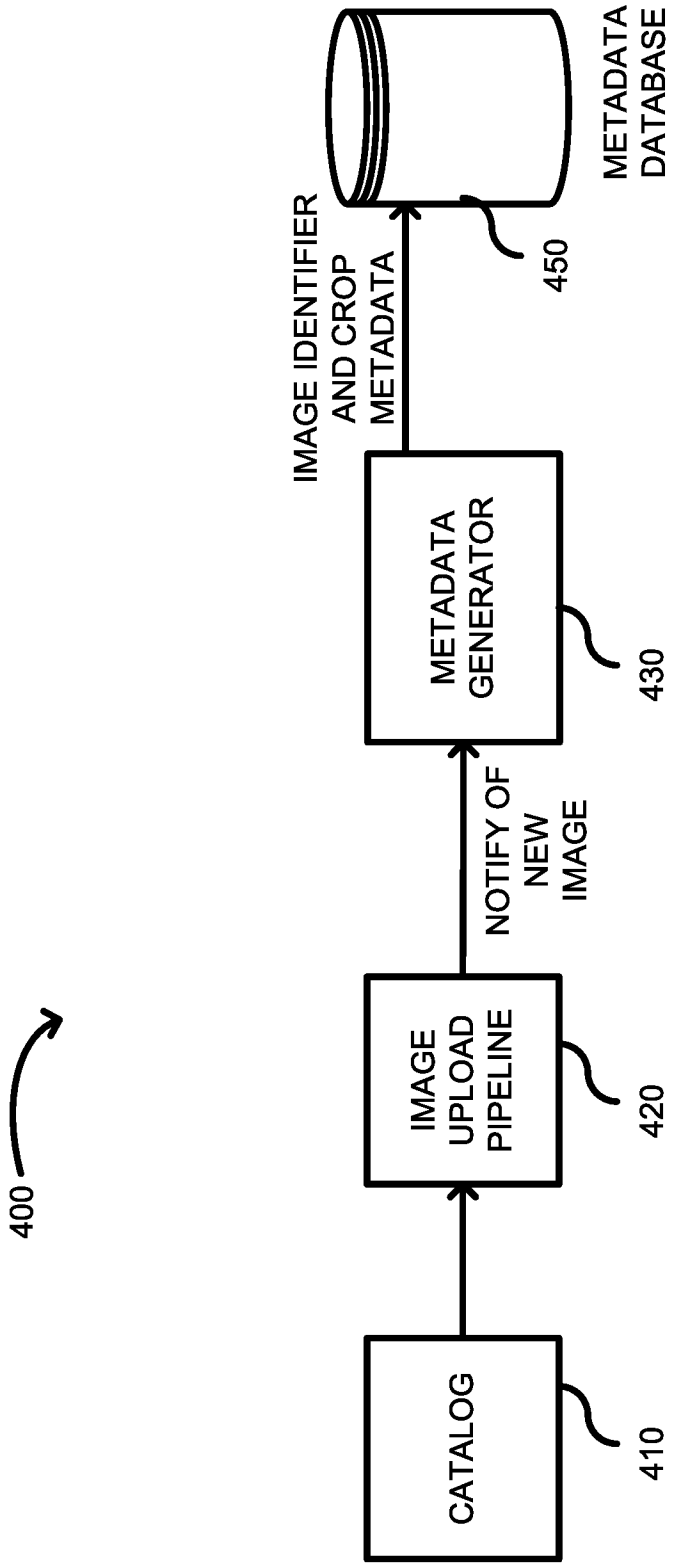

400
CATALOG
410
IMAGE UPLOAD PIPELINE
420
NOTIFY OF NEW IMAGE
METADATA GENERATOR
430
IMAGE IDENTIFIER AND CROP METADATA
450
METADATA DATABASE

FIG. 4

FIG. 5

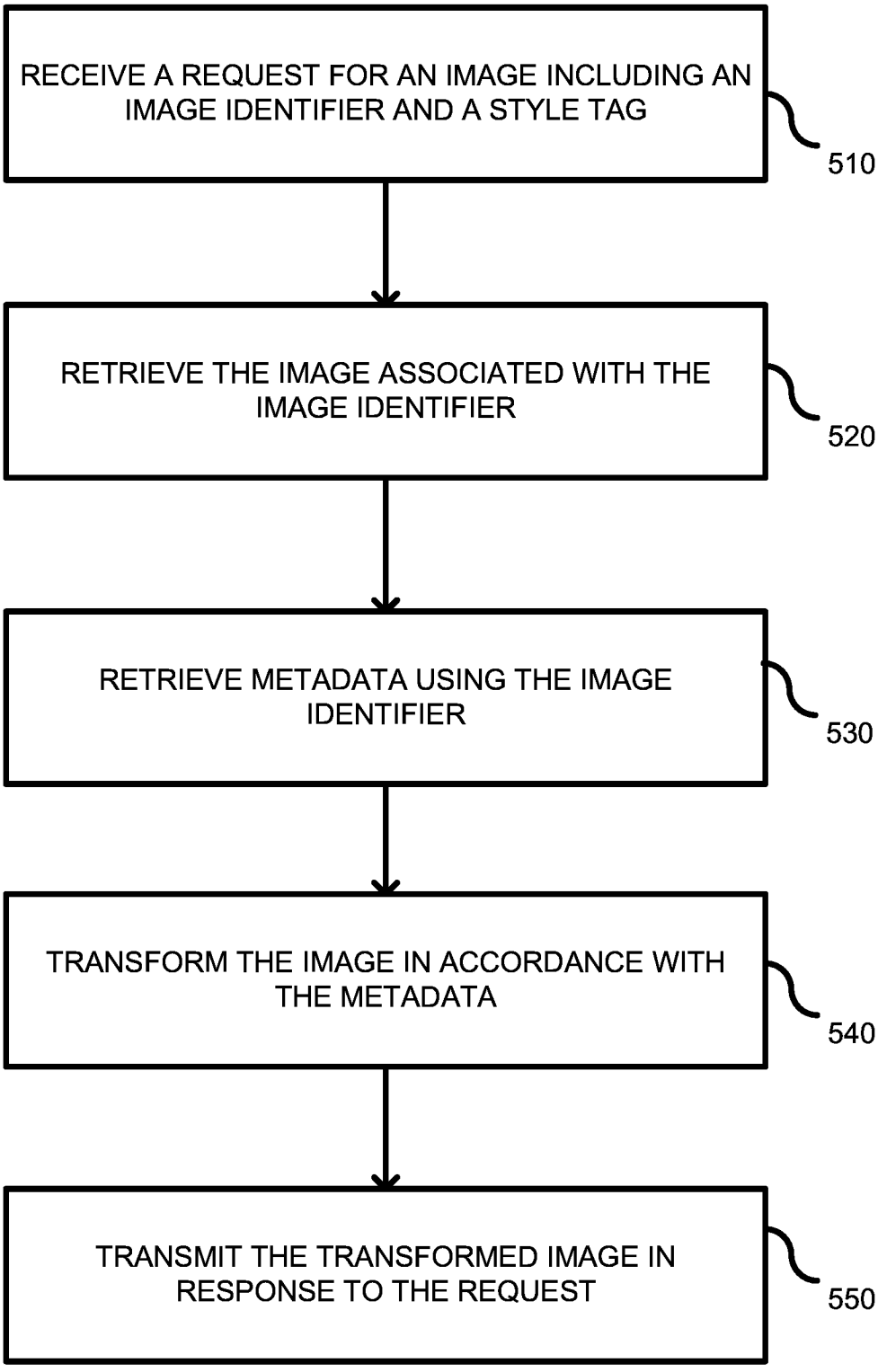

RECEIVE A REQUEST FOR AN IMAGE INCLUDING AN IMAGE IDENTIFIER AND A STYLE TAG
510
RETRIEVE THE IMAGE ASSOCIATED WITH THE IMAGE IDENTIFIER
520
RETRIEVE METADATA USING THE IMAGE IDENTIFIER
530
TRANSFORM THE IMAGE IN ACCORDANCE WITH THE METADATA
540
TRANSMIT THE TRANSFORMED IMAGE IN RESPONSE TO THE REQUEST
550

FIG. 6

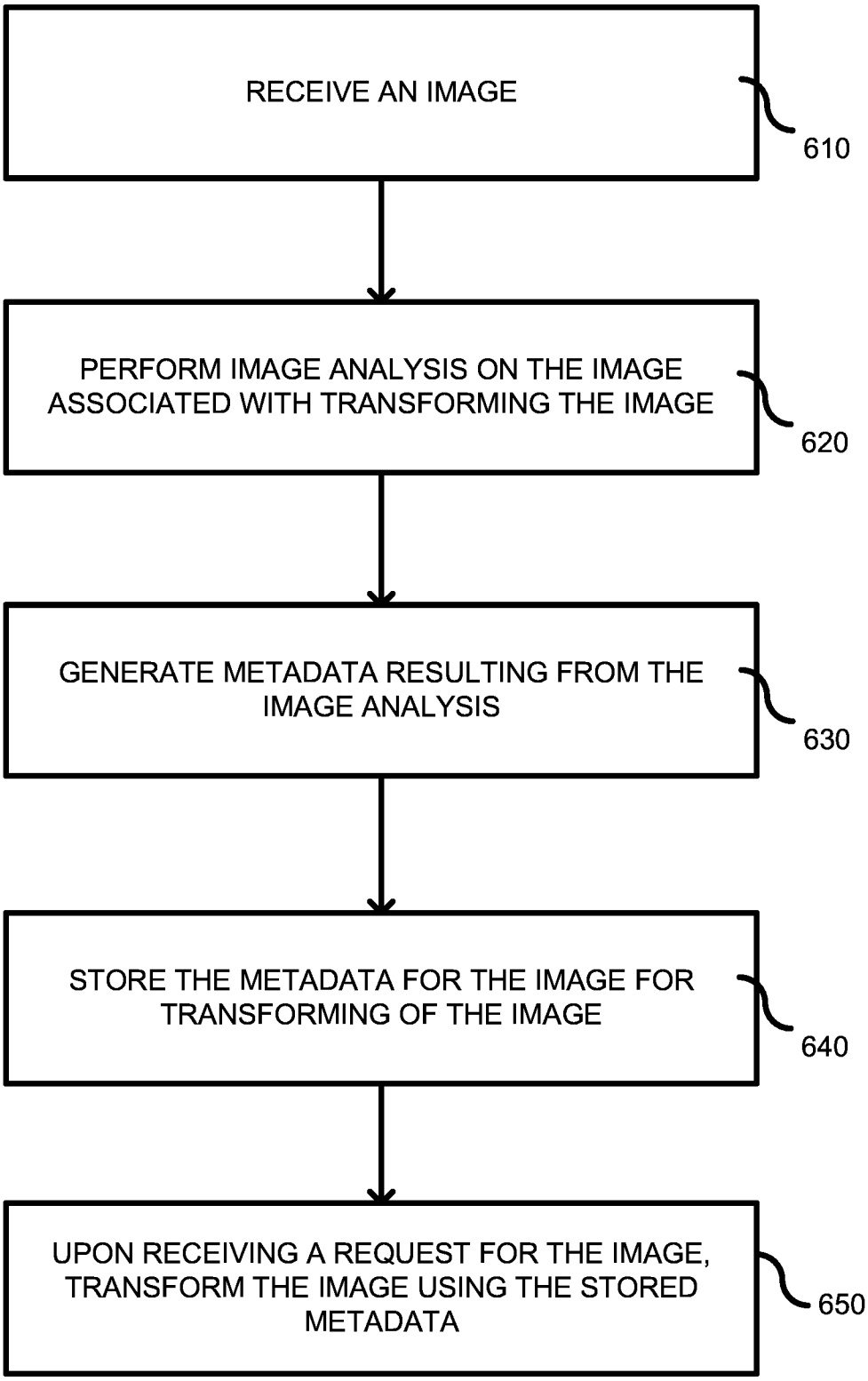
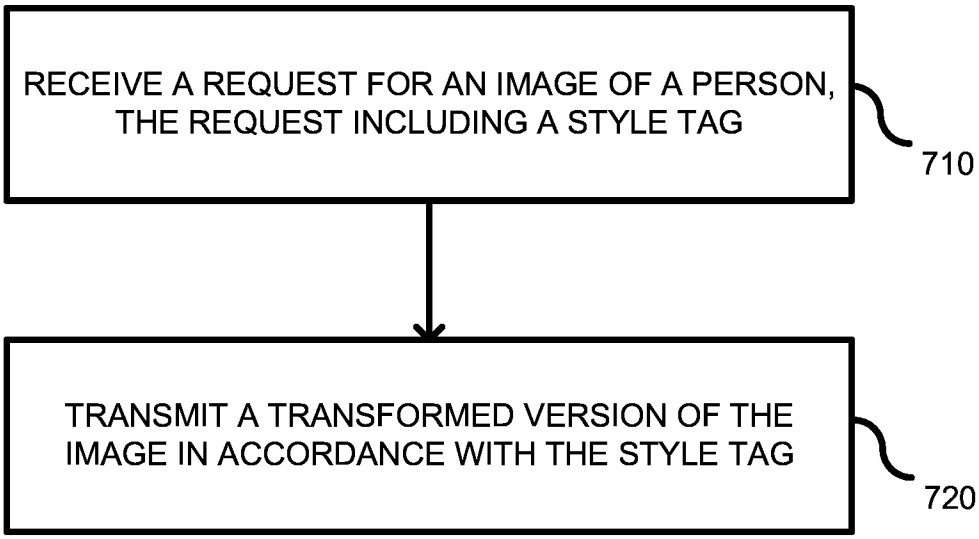

RECEIVE AN IMAGE
610
PERFORM IMAGE ANALYSIS ON THE IMAGE ASSOCIATED WITH TRANSFORMING THE IMAGE
620
GENERATE METADATA RESULTING FROM THE IMAGE ANALYSIS
630
STORE THE METADATA FOR THE IMAGE FOR TRANSFORMING OF THE IMAGE
640
UPON RECEIVING A REQUEST FOR THE IMAGE, TRANSFORM THE IMAGE USING THE STORED METADATA
650

FIG. 7


RECEIVE A REQUEST FOR AN IMAGE OF A PERSON, THE REQUEST INCLUDING A STYLE TAG
710
TRANSMIT A TRANSFORMED VERSION OF THE IMAGE IN ACCORDANCE WITH THE STYLE TAG
720

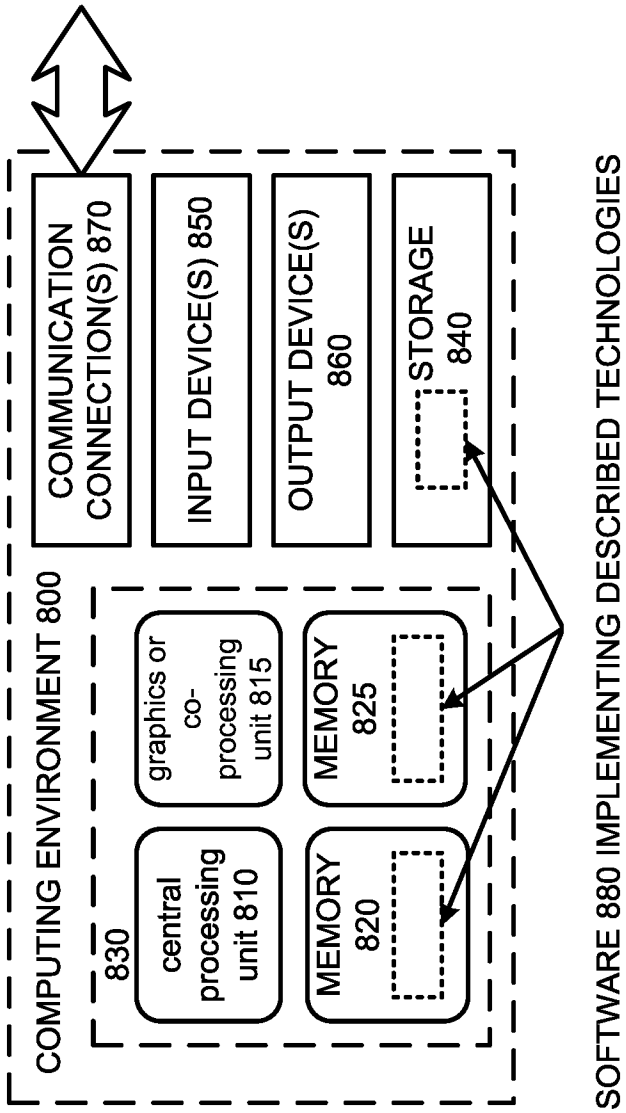

COMPUTING ENVIRONMENT 800
830
central processing unit 810
graphics or co-processing unit 815
MEMORY 820
MEMORY 825
COMMUNICATION CONNECTION(S) 870
INPUT DEVICE(S) 850
OUTPUT DEVICE(S) 860
STORAGE 840
SOFTWARE 880 IMPLEMENTING DESCRIBED TECHNOLOGIES

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/038949

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F16/583 G06K9/00
ADD.

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)
G06F G06K

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018/181593 A1 (RANZINGER MIKE [US] ET AL) 28 June 2018 (2018-06-28) abstract paragraph [0028] paragraph [0034] - paragraph [0035] paragraph [0051] - paragraph [0059] paragraph [0068] - paragraph [0070] paragraph [0075] - paragraph [0086] paragraph [0111] - paragraph [0116] paragraph [0132] -----	1-15
X	US 7 734 058 B1 (ADAMS CHERYL [US] ET AL) 8 June 2010 (2010-06-08) abstract column 2, line 7 - column 2, line 36 column 3, line 9 - column 3, line 40 column 4, line 11 - column 4, line 67 column 5, line 25 - column 6, line 10 ----- -/--	1-15

[X] Further documents are listed in the continuation of Box C.

[X] 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

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 October 2021	12/10/2021

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Boyadzhiev, Yavor

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2021/038949

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	KARMAKAR DHIMAN ET AL: "Face Recognition using Face-Autocropping and Facial Feature Points Extraction", PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON PERCEPTION AND MACHINE INTELLIGENCE, PERMIN '15, 1 January 2015 (2015-01-01), pages 116-122, XP055846757, New York, New York, USA DOI: 10.1145/2708463.2709056 ISBN: 978-1-4503-2002-3 Retrieved from the Internet: URL:http://dx.doi.org/10.1145/2708463.2709056> abstract page 116, right-hand column, line 35 - page 117, left-hand column, line 21 page 117, right-hand column, line 64 - page 118, left-hand column, line 21 -----	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2021/038949

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2018181593 A1	28-06-2018	NONE	
US 7734058 B1	08-06-2010	NONE	