March 10, 2015

Pipeline To Generate Training Data For Image Recognition

Rahul Garg

Sven Gowal

Pipeline To Generate Training Data For Image Recognition

Abstract: Image recognition programs require large sets of training data to produce accurate results. Human workers may categorize training sets that programs may use as training data to learn how to recognize objects. To increase the efficiency of the workers, it is proposed to break the categorization down into multiple steps in a pipeline. Different groups of workers will provide input at different stages of the pipeline, and the input from one group of workers will be passed to another group of workers. Breaking the categorization down into smaller tasks may increase the efficiency of the workers.

Image recognition programs require large sets of training data to produce accurate results. Human workers may categorize training sets that programs may use as training data to learn how to recognize objects. To increase the efficiency of the workers, it is proposed to break the categorization down into multiple steps in a pipeline. One group of workers may answer a first question with respect to a set of training data, and data based on their answers may be fed to a second group of workers. The second group of workers may answer a second question with respect to the data fed from the first workers, and so on until a final set of training data is generated for the image processing program. Breaking the categorization down into smaller tasks may increase the efficiency of the workers by allowing the workers to specialize and continually work within a same interface.

FIG. 1 shows a general pipeline for generating a training set. The pipeline may break the process of generating the training set down into multiple microtasks 104, 108, 114. A microtask 104, 108 preceding a given microtask may be considered a supported preceding task. A first task 104 may be considered a beginning task. Human workers performing each microtask 104, 108, 114 may answer a question based on data and/or answers from a preceding microtask, and their answers 106, 110, 112 may be fed to the next microtask 108, 114 to generate a new question.

In this example, pipeline details 102, which may be instructions for generating the training set, may be fed into microtask 1 104. The instructions may cause the workers

performing microtask 1 104 to acquire data, such as capturing video, photograph, or audio data. The acquired data may then be fed to microtask 2 108 as an answer.

The workers performing microtask 2 may then be asked whether the data fit within a predetermined category, or asked to assign the data to one of multiple predetermined categories. The data that fit within a predetermined category may then be fed to a next microtask as an answer 110, or data that fit within different categories may be fed to different microtasks as different answers. Microtasks may also include selecting portions of images that meet the predetermined category. A last microtask, shown as microtask N 114, may provide a final answer to provide information regarding the data, and finalize the training set for an image recognition program to learn from.

FIG. 2 shows steps performed within a single microtask, such as any of the microtasks 104, 108, 114 shown in FIG. 1. As shown in FIG. 2, an answer 202 from a preceding microtask may be fed to a back-end inserter 204. The back-end inserter 204 may be included on a server remote from the workers. The back-end inserter 204 may take in an answer from a previous microtask in the pipeline shown in FIG. 1 and generate question data from the answer. A supported preceding task is a task whose answer can be converted to question data by the back-end inserter 204. The back-end inserter 204 may present a question to a worker. The back-end inserter 204 may provide question data 206 to a front-end 208. The front-end 208 may be included on a computer used directly by the worker, and may include a web browser.

The front-end 208 may take in question data, such as images, video, audio, and/or instructions to the worker, and display or render the question to the worker. The front-end 208 may take in the worker's response and send the worker's response back along with the original

question data to a back-end listener 212. The back-end listener 212 may be included on the server.

The front-end 208 may include a web browser or other user interface presenting data, such as a video, image, or audio, as well as a question, based on the question data 206 received from the back-end inserter 204, to a worker. The worker may provide input into the front-end by answering the question. The front-end 208 may provide the question data and worker response 210 to a back-end listener 212.

The back-end listener 212 may take in the question data and the worker's response and construct the answer from the microtask to be passed on to the next step in the pipeline. The answer could also be logged to a backend database. The back-end listener 212 may then determine whether to feed the question data and worker response 210 to the next microtask and if so, provide the answer 214 based on the question data and worker response 210 to the next microtask.

FIG. 3 shows a pipeline for generating a training set of "thumbs-up" gestures including an indication of which hand made the thumbs-up gesture. The pipeline of FIG. 3 is a specific example of the general pipeline of FIG. 1. The pipeline of FIG. 2 may include capturing videos of thumbs-up gestures 304, selecting frames that include the thumbs-up gesture 308, drawing a bounding box 310 around the parts of the frames that include the thumbs up gesture, and selecting left hand or right hand 314 to indicate whether the thumbs-up gesture is made with the right hand or the left hand.

As shown in FIG. 3, pipeline details 302 may be fed to the capture video microtask 304. The pipeline details 302 may include instructions for workers to capture video of thumbs up gestures. Referring to FIG. 2, the back-end inserter 204 of the capture video microtask 304 may

send question data 206 to the front-end 208 instructing the front-end 208 to ask the worker to take videos of himself or herself making a thumbs up gesture, with different hands. To change the background, which helps machine learning algorithms to learn, the worker may be encouraged to move his or her hand while performing the gesture, such as by following a dot on a screen. The videos may then be fed to the back-end listener 212 as the question data and worker response 210. The back-end listener 212 may then feed the videos as an answer 214/306 to the contains gesture microtask 308. The back-end listener 212 may feed the raw videos to the contains gesture microtask 308, or may break the video down into frames with a lower frequency, such as one or two frames per second, and send the frames to the contains gesture microtask 308, reducing the amount of data to be sent.

The contains gesture microtask 308 may receive the video or frames from the capture video microtask 304 as the answer 306. The workers performing the contains gesture microtask 308 may be different than the workers that performed the capture video microtask. The back-end inserter 204 of the contains gesture microtask 308 may send the videos or frames to the front-end 208 as question data 206. The front-end 208 may ask the worker to categorize the videos or frames as either containing a thumbs-up gesture or not containing a thumbs-up gesture.

FIG. 4 shows an example of a front-end interface 402 for the contains gesture microtask 308. The front-end interface 402 may be generated by a web browser. In this example, the front-end interface 402 may include instructions 404. The instructions 404 in this example are, “Select the images that contain the shown gesture,” with a picture of a thumbs-up gesture. The front-end interface 402 may include multiple frames 406 or photographs, some of which include a thumbs-up gesture and some of which do not include a thumbs-up gesture.

In the example shown in FIG. 4, the worker may quickly choose selected frames 410 which include the thumbs-up gesture by surrounding them with a rectangle or polygon using a mouse gesture, and the non-selected frames 408 which do not include the thumbs-up gesture may be the frames 406 that were not surrounded with a rectangle or polygon. In another example, the worker may select frames 410 the include the thumbs-up gesture by clicking on the frames 410 that do include the thumbs-up gesture. After the worker has selected and/or indicated which frames 406 include the thumbs-up gesture, the worker may click a submit button 412.

Returning to FIG. 2, the front-end 208 of the contains gesture microtask 308 may provide an indication of which frames the worker identified as including the thumbs-up gesture to the back-end listener 212 as the question data and worker response 210. If the worker did not identify any of the frames 406 as including the thumbs-up gesture, then the contains gesture microtask 308 may not send any data and/or answer 309 to the draw bounding box microtask 310 based on the answer 306 received from the capture video microtask 304. If the worker did identify at least one frame as including the thumbs-up gesture, then the contains gesture microtask 308 may send the selected frames to the draw bounding box microtask 310 as the answer 214/309.

The back-end inserter 204 of the draw bounding box microtask 310 may receive, as an answer 202/310, the selected frames 410 that were identified as including the thumbs-up gesture. The back-end inserter 204 may send, to the front-end 208 as question data 206, the selected frames 410 with an instruction to the workers to draw a bounding box around the portion of each frame that includes the thumbs-up gesture. The workers performing the draw bounding box microtask 310 may be different than the workers performing the capture video microtask 304 and/or contains gesture microtask 308, increasing the efficiency and productivity of each worker.

FIG. 5 shows a front-end interface 502 sent to the worker by the draw bounding box microtask 310. The front-end interface 502 may be generated by a web browser. In this example, the front-end interface 502 may include instructions 504. The instructions in this example are, "Draw a tight bounding box around the hand in the image. If there's no hand in the image, click 'No Hand' to skip to the next image. Start by clicking on the left edge of hand as shown in the image to the right," with a picture of a thumbs-up gesture. The front-end interface 502 may include an image from the selected frames 410, and the front-end interface 502 may allow the worker to draw a rectangle around the thumbs-up gesture using, for example, a mouse.

The front-end interface 502 may include buttons 508. The buttons 508 may include an undo button allowing the worker to erase the bounding box and create a new bounding box, a submit button allowing the worker to submit the drawn rectangle and/or bounding box, and a no/blurry hand button allowing the worker to indicate that the image 506 did not include a thumbs-up gesture.

The front-end 208 of the draw bounding box microtask 310 may send coordinates of the rectangles and/or bounding boxes of the submitted bounding boxes to the back-end listener 212 as the question data and worker response 210, or may send the smaller images themselves that were bounded by the rectangles and/or bounding boxes to the back-end listener 212 as the question data and worker response 210. The back-end listener 212 of the draw bounding box microtask 310 may send the bounded images of the thumbs-up gestures that were submitted by the worker to the select left/right hand microtask as answers 214, 312.

The select left/right hand microtask 314 may ask workers whether the thumbs-up gesture is made using the right hand or the left hand. The back-end inserter 204 of the select left/right hand microtask 314 may send the bounded images, and an instruction to indicate whether the

images were made using the left hand or the right hand, to the front-end 208 as question data 206. The front-end 208 of the select left/right hand microtask 314 may present the images to the workers and request the workers to indicate whether the thumb-up images were made using the left hand or the right hand. The workers performing the select left/right hand microtask may be different than the workers performing the capture video microtask, contains gesture microtask, and/or draw bounding box microtask 310, increasing the efficiency and productivity of each worker.

FIG. 6 shows a front-end interface 600 presented by the select left/right hand microtask 314. The front-end interface 600 may be generated by a web browser. The front-end interface 600 may include instructions to, “Select whether the image contains right hand or left hand.” The front-end interface 600 may also include an image 604. The image 604 may include a portion of a selected frame 410 that was bounded during the draw bounding box microtask 310.

The front-end interface 600 may include a first button 606 for the worker to indicate that the image 604 includes a left hand, a second button 608 for the worker to indicate that the image 604 includes a right hand, and a null button 610 for the worker to indicate that there is no hand in the image.

The front-end 208 of the select left/right hand microtask 314 may provide, to the back-end listener 212, the indications of left hand, right hand, or no hand in the image 604, as question data and worker response 210. The back end listener 212 may then provide, to an image processing program as an answer 214, 316, the bounded portions of the selected images along with an indication that the images include thumbs-up gestures and whether the gestures were made using a left hand or a right hand.

Multiple workers may perform each of the microtasks. Each worker may perform the microtask on different portions of the data. However, some or all of the identical data may be sent to multiple workers, and if some workers give different answers for the same data, the answers provided by a majority of the workers may be used.

The framework and/or pipeline described above may be used for different types of gestures and/or training sets for image recognition. Images and/or requests to capture images may be broken down into the multiple steps described above and distributed to multiple workers within a pipeline. Asking each worker to perform the same step within the pipeline repeatedly increases the efficiency and productivity of the workers, while minimizing human errors.

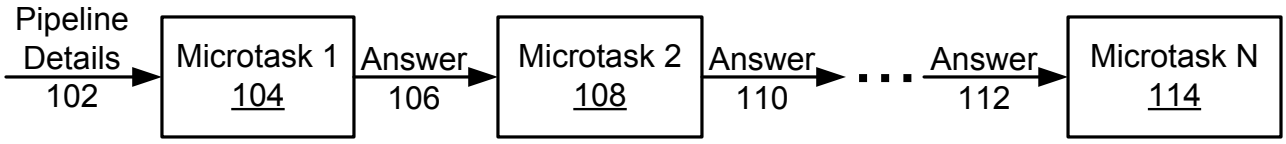
Pipeline Details
102
Microtask 1
104
Answer
106
Microtask 2
108
Answer
110
Answer
112
Microtask N
114

FIG. 1

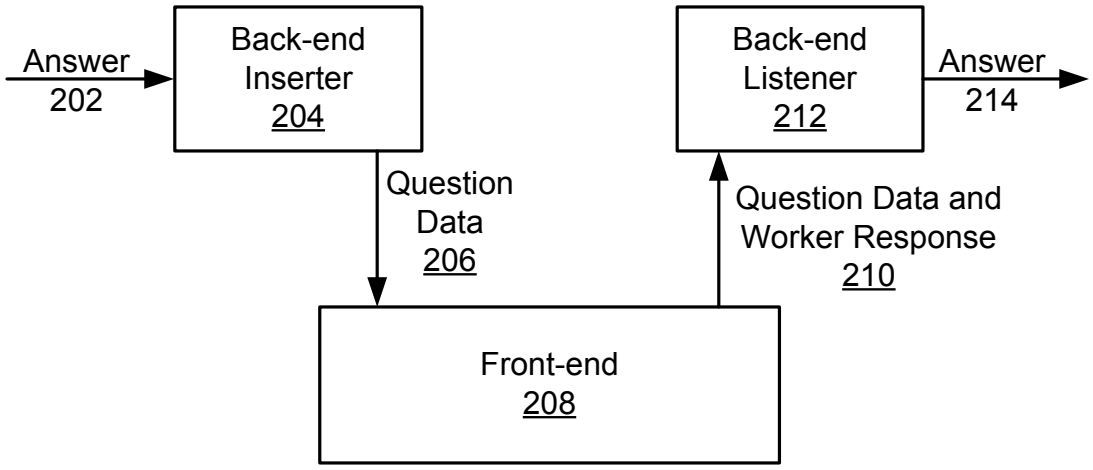
Answer
202
Back-end Inserter
204
Question Data
206
Front-end
208
Question Data and Worker Response
210
Back-end Listener
212
Answer
214

FIG. 2

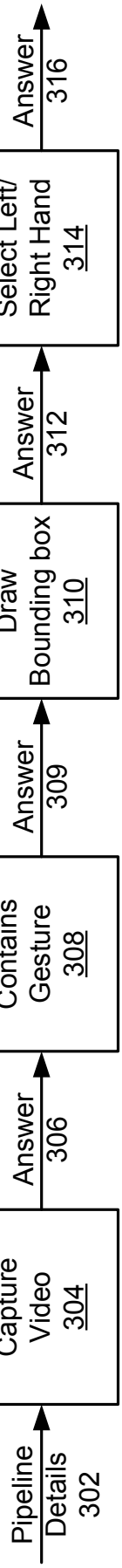
Pipeline Details 302
Capture Video 304
Answer 306
Contains Gesture 308
Answer 309
Draw Bounding box 310
Answer 312
Select Left/ Right Hand 314
Answer 316

FIG. 3

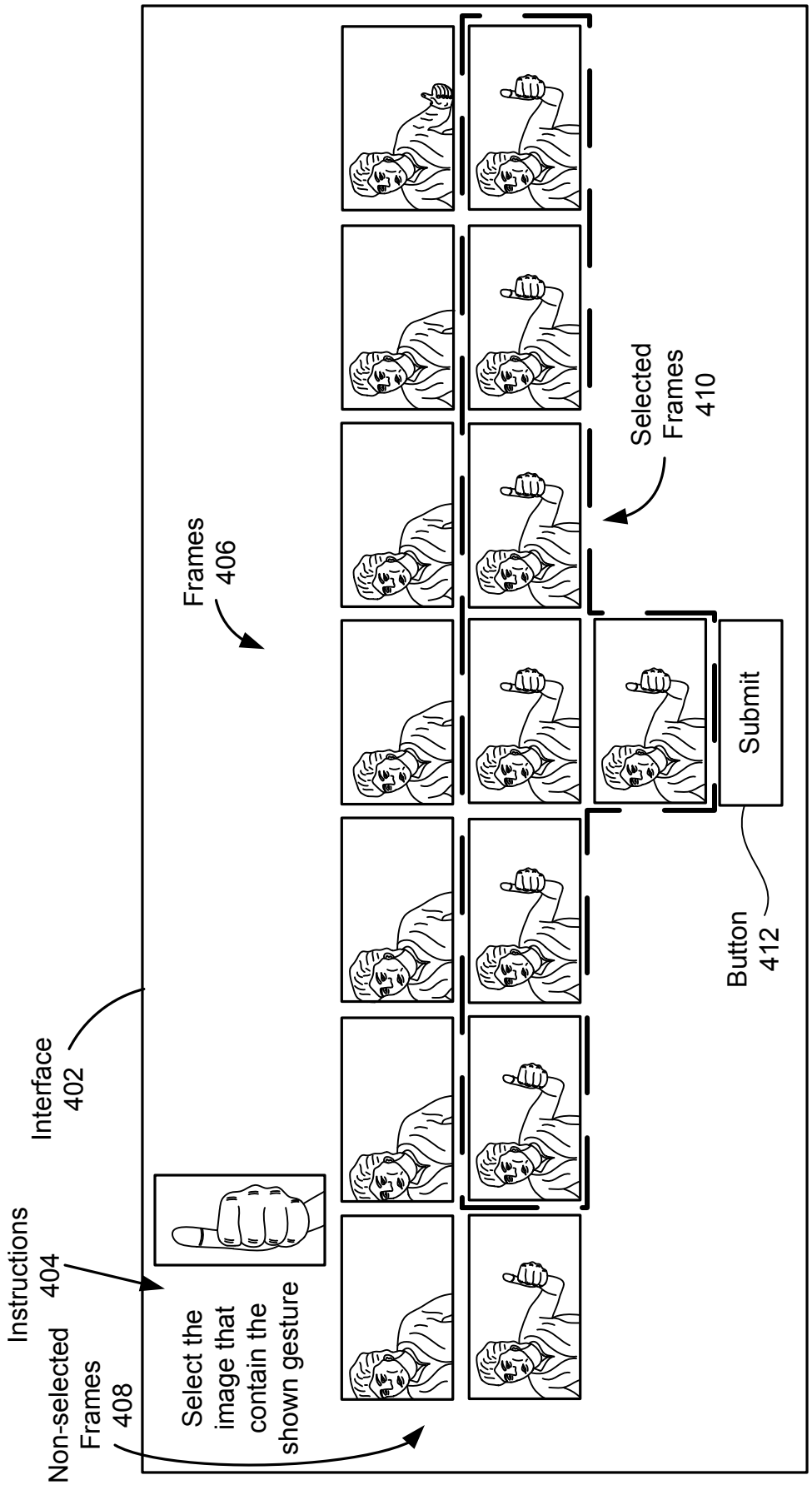
Interface
402
Instructions
404
Select the
image that
contain the
shown gesture
Non-selected
Frames
408
Frames
406
Selected
Frames
410
Button
412
Submit

FIG. 4

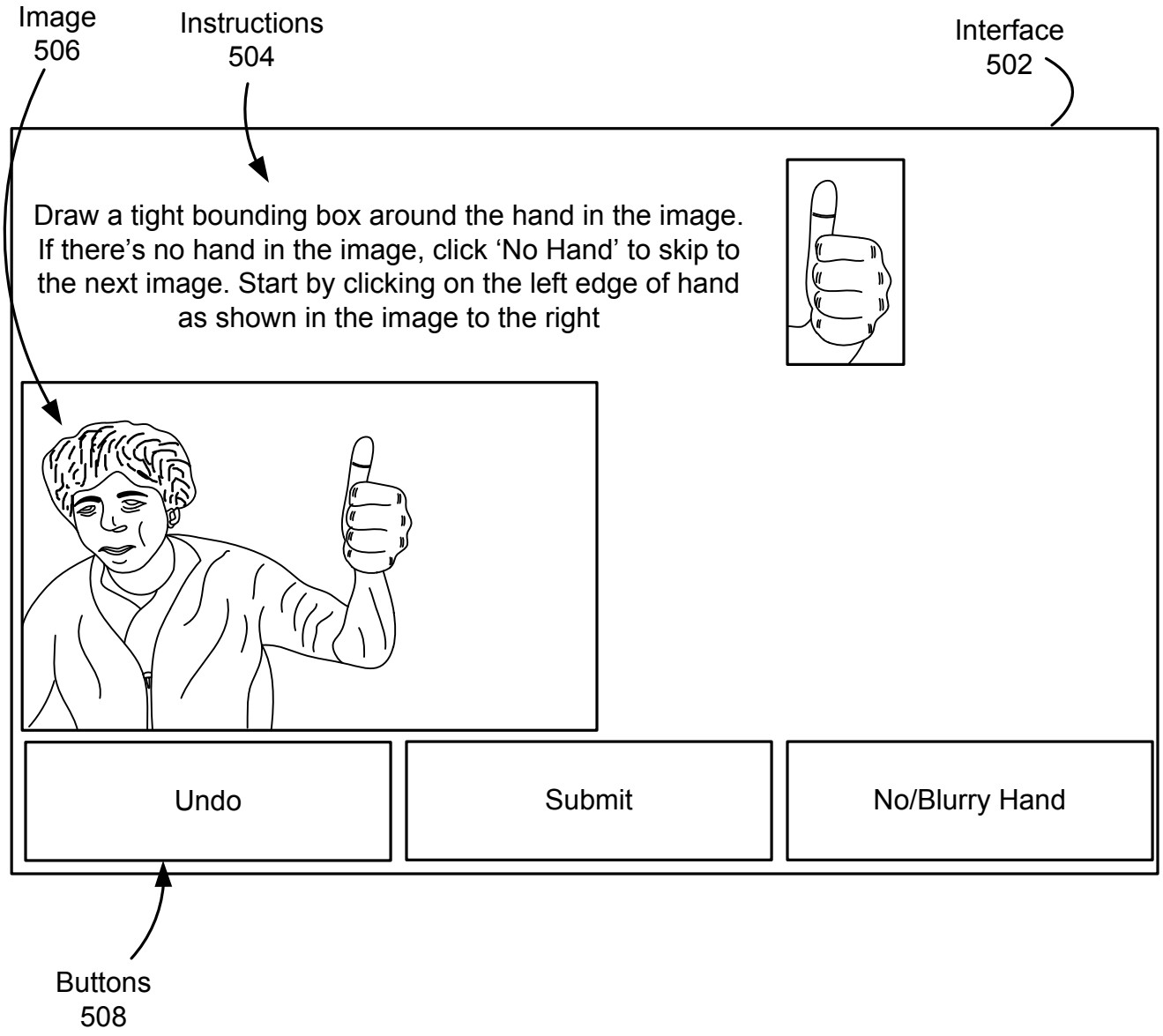
Image
506
Instructions
504
Interface
502
Draw a tight bounding box around the hand in the image. If there's no hand in the image, click 'No Hand' to skip to the next image. Start by clicking on the left edge of hand as shown in the image to the right
Undo
Submit
No/Blurry Hand
Buttons
508

FIG. 5

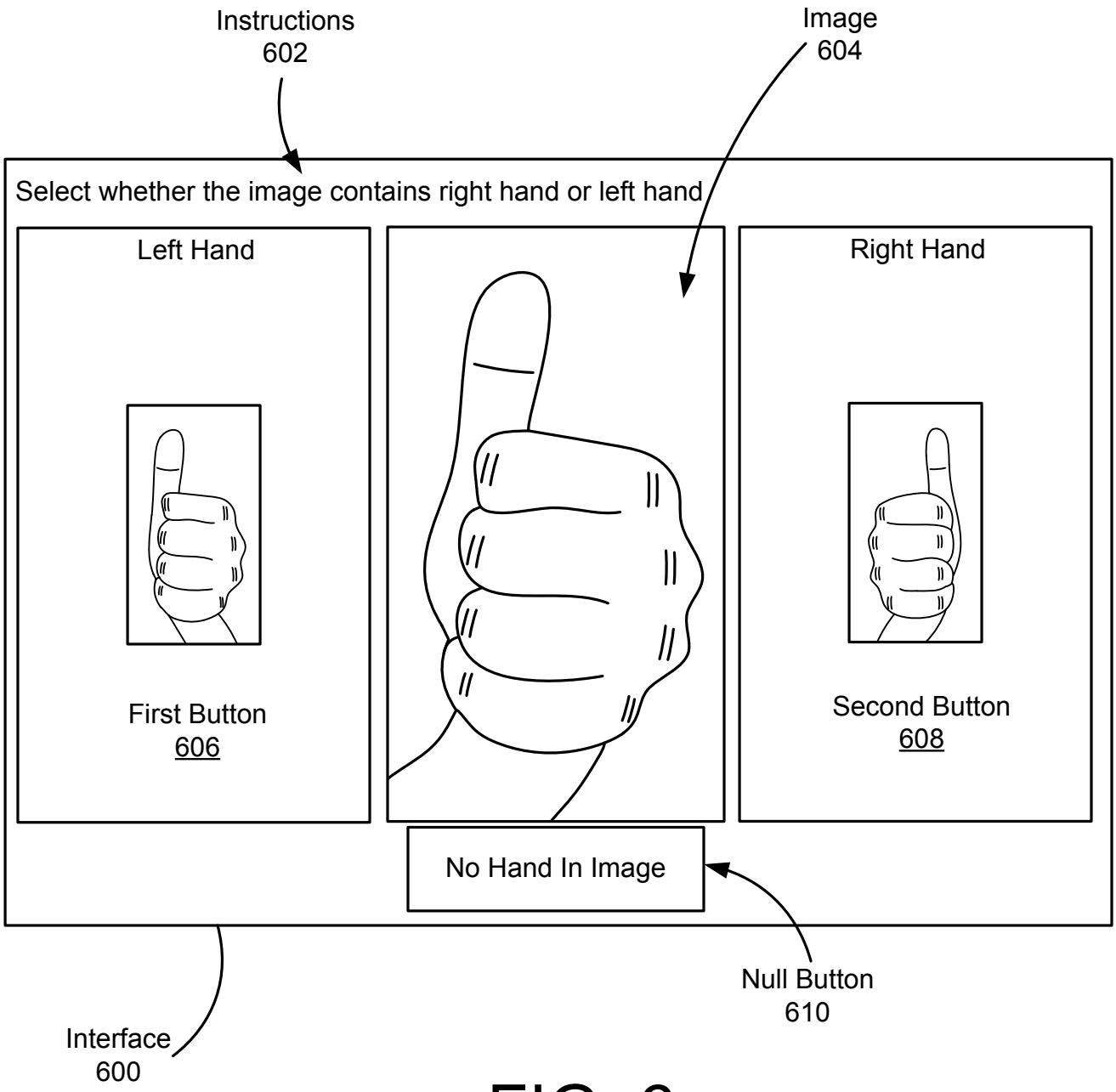
Instructions
602
Image
604
Select whether the image contains right hand or left hand
Left Hand
Right Hand
First Button
606
Second Button
608
No Hand In Image
Null Button
610
Interface
600

FIG. 6