

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

UNITED STATES PATENT NO. 8,447,132 ('132 Patent)

INFRINGEMENT CHART FOR CLAIMS 1, 5, 6, 13, 21, 23, 33¹

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
Claim 1	
[1a.] A method comprising; ⁴	The '132 Patent Accused Products defined in Section II of Qualcomm Incorporated's Patent L.R. 3.1 and 3.2 Disclosures include one or more of a central processing unit, a graphics processing unit, and an image processor. For example, the iPhone 7 and 7 Plus include an Image Signal Processor (ISP) as well as the A10 fusion processor (which includes a CPU and a GPU), any of which perform image signal processing. See, e.g., http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-7-teardown/ . Additionally, the iPhone 8, 8 Plus, and X include an Image Signal Processor (ISP) and the A11 Bionic processor (which includes a CPU and a GPU), any of which perform image signal processing. See, e.g., http://techinsights.com/about-techinsights/overview/blog/apple-iphone-8-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-x-teardown/ . The iPhone 5, 5s, 5c, 6, 6 Plus, 6s, and SE each also include processors such as a central processing unit (CPU) and a graphics processing unit (GPU), which can each perform image signal processing. See, e.g., http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-5-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-5s-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-5c-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-6-and-iphone-6-plus-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/inside-the-iphone-6s/ ;


¹ The infringement contentions provided herein are based on information obtained to date and may not be exhaustive. Qualcomm's investigation of Apple's infringement is ongoing. Qualcomm reserves the right to supplement and/or amend these disclosures to identify additional Asserted Claims (P.R. 3-1(a)), to identify additional accused products (P.R. 3-1(b)), and to further identify where each element of each Asserted Claim is found in each of the '132 Patent Accused Products (P.R. 3-1(c)), including on the basis of discovery obtained from Apple and from third-parties during the course of this litigation.

² All infringement contentions set forth herein for any independent patent claims are hereby incorporated by reference into the infringement contentions alleged for any dependent patent claims that depend on such independent claims, as if fully set forth therein.

³ The '132 Patent Accused Products and associated exhibits discussed and/or cited for any claim herein are representative in all material respects of all other '132 Patent Accused Products identified for that claim (e.g., although various Apple devices may have immaterial differences in their hardware, firmware, and/or software configuration, the mappings of claim limitations to specific aspects of the Apple devices are believed to be illustrative of all '132 Patent Accused Products).

⁴ Qualcomm's inclusion of any claim preamble in this claim chart should not be interpreted as an admission that the preamble is limiting. Qualcomm reserves the right to take the position that the claim preambles are limiting or not limiting on a claim-by-claim basis.

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-se-teardown/. The iPad (3rd generation), iPad (4th generation), iPad Air, iPad Air 2, iPad (2017), iPad Mini, iPad Mini 2, iPad Mini 3, iPad Mini 4, iPad Pro (1st generation), and iPad Pro (2nd generation) each also include processors such as a central processing unit (CPU) and a graphics processing unit (GPU), each of which can perform image signal processing. See, e.g., https://support.apple.com/kb/SP647?locale=en_US; https://support.apple.com/kb/SP662?locale=en_US; https://support.apple.com/kb/SP692?locale=en_US; https://support.apple.com/kb/SP708?viewlocale=en_US&locale=en_US; https://www.apple.com/ipad-9.7/specs/; https://support.apple.com/kb/SP661?locale=en_US; https://support.apple.com/kb/SP693?locale=en_US; https://support.apple.com/kb/SP709?locale=en_US&viewlocale=en_US; https://www.apple.com/ipad-mini-4/specs/.</p>  <p>But behind it all is the brains of the camera: the image signal processor. And this enables so much of the unique quality that we get in the pictures we take with iPhone, and the ISP and the new chip in the iPhone 7 is twice the throughput our previous versions.</p> <p>Apple Special Event, September 7, 2016 at 1:03:30, What does this ISP (Image Signal Processor) do?") available at https://www.apple.com/apple-events/september-2016/.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>For example, the iPhone 7 Plus has an ISP. (Describing the iPhone 7 Plus) “The TSVs are used to connect the CIS chip to an underlying image signal processor (ISP), which is not to be confused with the complementary ISP functions embedded within the A10 chip.</p> <p>http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-7-teardown/</p> <p>For example, the iPhone 8 and 8Plus have “new Apple-designed image signal processors (ISP) that help improve autofocus (AF) performance in low-light, and feature pixel processing for enhanced sharpness and texture. First time, the ISP on iPhone’s application processor (AP) features hardware-enabled multi-band noise reduction. Of course, this new AP functionality complements the processing done on the stacked chip image sensors’ companion ISPs.</p> <p>http://techinsights.com/about-techinsights/overview/blog/apple-iphone-8-teardown/#03</p> <p>A new color filter, deeper pixels and an improved Apple-designed image signal processor delivers advanced pixel processing, wide color capture, faster autofocus in low light and better HDR photos.”; “The A11 Bionic CPU handles world tracking, scene recognition and the GPU enables incredible graphics at 60fps, while the image signal processor does real-time lighting estimation.</p> <p>https://www.apple.com/newsroom/2017/09/the-future-is-here-iphone-x/</p> <p>The '132 Patent Accused Products perform the method of claim 1, such as when Apple tests the '132 Patent Accused Products. <i>See</i> [1b]-[1e]. The method is performed by one or more of a central processing unit, a graphics processing unit, and an image processor.</p>
[1b.] determining whether a first portion of digital image data represents a physical object of a predetermined type;	<p>The '132 Patent Accused Products determine whether a first portion of digital image data represents a physical object of a predetermined type.</p> <p><i>See</i> [21b].</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found³ (P.R. 3-1(b)-(c))
[1c.] determining a correction to apply to the first portion of the digital image data, based on a determination that the first portion of the digital image data represents a physical object of the predetermined type, wherein the determined correction is matched to the predetermined type;	The '132 Patent Accused Products determine a correction to apply to the first portion of the digital image data, based on a determination that the first portion of the digital image data represents a physical object of the predetermined type, wherein the determined correction is matched to the predetermined type. <i>See</i> [21b]-[21c].
[1d.] applying the determined correction to the first portion of the digital image data to enhance a visual characteristic of the first portion of the digital image data, by applying a first amount of the correction to the first portion of the digital image data; and	The '132 Patent Accused Products apply the determined correction to the first portion of the digital image data to enhance a visual characteristic of the first portion of the digital image data, by applying a first amount of the correction to the first portion of the digital image data. <i>See</i> [21b]-[21c].
[1e.] applying a second amount of the correction to a second portion of the digital image data, wherein the first amount differs from the second amount, and wherein the first amount corresponds to a physical object of the predetermined type.	The '132 Patent Accused Products apply a second amount of the correction to a second portion of the digital image data, wherein the first amount differs from the second amount, and wherein the first amount corresponds to a physical object of the predetermined type. <i>See</i> [21d].
Claim 5	
[5a.] A method as recited in claim 1,	The '132 Patent Accused Products perform the method of claim 1. <i>See</i> [1a]-[1e].

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found³ (P.R. 3-1(b)-(c))
<p>[5b.] wherein applying the determined correction to the first portion of the digital image data comprises applying a first amount of a parameter to the first portion to enhance the visual characteristic, wherein the first amount of the parameter is different from a second amount of the parameter applied to a second portion of the digital image data, resulting in a first portion of a digital image that corresponds to the first portion of digital image data being brighter than if the second amount of the parameter were applied to both the first portion and the second portion of the digital image data.</p>	<p>The '132 Patent Accused Products apply a first amount of a parameter to the first portion to enhance the visual characteristic, wherein the first amount of the parameter is different from a second amount of the parameter applied to a second portion of the digital image data, resulting in a first portion of a digital image that corresponds to the first portion of digital image data being brighter than if the second amount of the parameter were applied to both the first portion and the second portion of the digital image data.</p> <p><i>See</i> [21b]-[21d].</p> <p><i>See, e.g.,:</i></p> <p>Core Image can analyze and find human faces in an image. It performs face detection, not recognition. Face detection is the identification of rectangles that contain human face features, whereas face recognition is the identification of specific human faces (John, Mary, and so on). After Core Image detects a face, it can provide information about face features, such as eye and mouth positions. It can also track the position an identified face in a video.</p> <p>Knowing where the faces are in an image lets you perform other operations, such as cropping or adjusting the image quality of the face (tone balance, red-eye correction and so on). You can also perform other interesting operations on the faces; for example:</p> <p>Anonymous Faces Filter Recipe shows how to apply a pixellate filter only to the faces in an image.</p> <p>White Vignette for Faces Filter Recipe shows how to place a vignette around a face.</p> <p>Core Image Programming Guide, Detecting Faces in Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html#apple_ref/doc/uid/TP30001185-CH8-SW1.</p> <p>The auto enhancement feature of Core Image analyzes an image for its histogram, face region contents, and metadata properties. It then returns an array of CIFilter objects whose input parameters are already set to values that will improve the analyzed image.</p> <p>Auto enhancement is available in iOS v5.0 and later and in OS X v10.8 and later.</p>

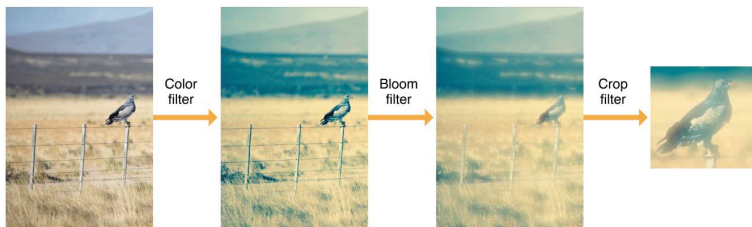
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))												
	<p>Core Image Programming Guide, Auto Enhancing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html.</p> <p>The autoAdjustmentFiltersWithOptions: method returns an array of options filters that you'll then want to chain together and apply to the analyzed image, as shown in Listing 3-1.</p> <p>Table 3-1 Filters that Core Image uses to enhance an image</p> <table border="1"> <thead> <tr> <th>Filter</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>CIRedEyeCorrection</td><td>Repairs red/amber/white eye due to camera flash</td></tr> <tr> <td>CIFaceBalance</td><td>Adjusts the color of a face to give pleasing skin tones</td></tr> <tr> <td>CIVibrance</td><td>Increases the saturation of an image without distorting the skin tones</td></tr> <tr> <td>CIToneCurve</td><td>Adjusts image contrast</td></tr> <tr> <td>CIHighlightShadowAdjust</td><td>Adjusts shadow details</td></tr> </tbody> </table> <p>Core Image Programming Guide, Using Auto Enhancement Filters, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html.</p> <p>Return Value An array of CIFilter instances preconfigured for correcting deficiencies in the supplied image.</p> <p>Apple Developer Documentation, autoAdjustFilters(options:), https://developer.apple.com/documentation/coreimage/ciimage/1437792-autoadjustmentfilters.</p>	Filter	Purpose	CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash	CIFaceBalance	Adjusts the color of a face to give pleasing skin tones	CIVibrance	Increases the saturation of an image without distorting the skin tones	CIToneCurve	Adjusts image contrast	CIHighlightShadowAdjust	Adjusts shadow details
Filter	Purpose												
CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash												
CIFaceBalance	Adjusts the color of a face to give pleasing skin tones												
CIVibrance	Increases the saturation of an image without distorting the skin tones												
CIToneCurve	Adjusts image contrast												
CIHighlightShadowAdjust	Adjusts shadow details												

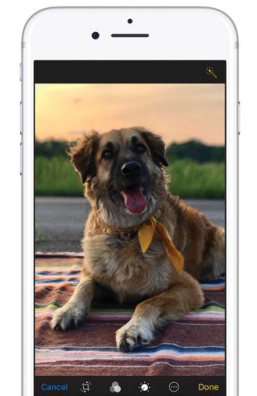
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Overview</p> <p>The CIFilterGenerator class provides methods for creating a CIFilter object by chaining together existing CIFilter objects to create complex effects. (A filter chain refers to the CIFilter objects that are connected in the CIFilterGenerator object.) The complex effect can be encapsulated as a CIFilterGenerator object and saved as a file so that it can be used again. The filter generator file contains an archived instance of all the CIFilter objects that are chained together.</p> <p>Any filter generator files that you copy to /Library/Graphics/Image Units/ are loaded when any of the loading methods provided by the CIPugin class are invoked. A CIFilterGenerator object is registered by its filename or, if present, by a class attribute that you supply in its description.</p> <p>You can create a CIFilterGenerator object programmatically, using the methods provided by the CIFilterGenerator class, or by using the editor view provided by Core Image.</p> <p>Core Image Documentation, CIFilterGenerator, https://developer.apple.com/documentation/coreimage/cifiltergenerator.</p> <p>sourceKey</p> <p>The key that specifies the source object. For example, if the source is the output image of a filter, pass the outputImage key. Pass nil if the source object is used directly.</p> <p>Core Image Documentation, connect(_ :withKey:to:withKey:), https://developer.apple.com/documentation/coreimage/cifiltergenerator/1438159-connect.</p> <p>Core Image filters process and produce Core Image images. A CIImage instance is an immutable object representing an image. These objects don't directly represent image bitmap data—instead, a CIImage object is a “recipe” for producing an image. One recipe might call for loading an image from a file; another might represent output from a filter, or from a chain of filters. Core Image performs these recipes only when you request that an image be rendered for display or output.</p> <p>Core Image Programming Guide, Processing Images,</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_tasks/ci_tasks.html.</p> <p>Chaining Filters for Complex Effects</p> <p>Every Core Image filter produces an output CIColor object, so you can use this object as input to another filter. For example, the sequence of filters illustrated in Figure 1-1 applies a color effect to an image, then adds a glow effect, and finally crops a section out of the result.</p> <p>Figure 1-1 Construct a filter chain by connecting filter inputs and outputs</p>  <p>Core Image Programming Guide, Processing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_tasks/ci_tasks.html.</p>

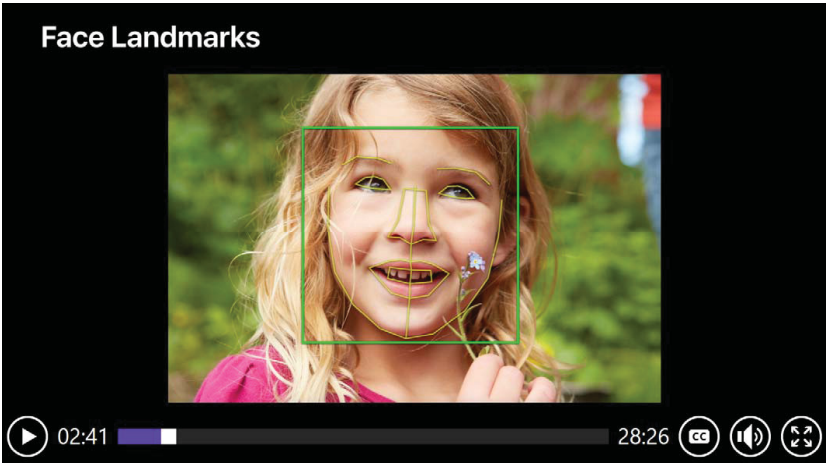
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Enhance instantly</p>  <p>Auto-enhance will instantly improve your picture's exposure, contrast, saturation, and other qualities. It's the fastest way to make your photo even more amazing.</p> <p>https://support.apple.com/en-us/ht205659</p>
Claim 6	

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
[6a.] A method as recited in claim 1,	The '132 Patent Accused Products perform the method of claim 1. <i>See</i> [1b]-[1e].
[6b.] wherein the predetermined type of physical object represents facial attributes.	<p>The '132 Patent Accused Products perform the method of claim 1, wherein the predetermined type of physical object represents facial attributes.</p> <p><i>See</i> [21b]-[21d].</p> <p><i>See, e.g.,</i>:</p> <p style="padding-left: 40px;">Getting Face and Face Feature Bounds</p> <p style="padding-left: 40px;">Face features include: left and right eye positions mouth position tracking ID and tracking frame count which Core Image uses to follow a face in a video segment (available in iOS v6.0 and later and in OS X v10.8 and later)</p> <p style="padding-left: 40px;">After you get an array of face features from a CIDetector object, you can loop through the array to examine the bounds of each face and each feature in the faces, as shown in Listing 2-2.</p> <p>Core Image Programming Guide, Detecting Faces in Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html#apple_ref/doc/uid/TP30001185-CH8-SW1.</p>

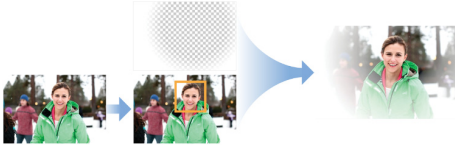
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div><p>Face Landmarks</p><p>WWDC 2017 - Session 506 - iOS, macOS, tvOS, Vision Framework: Building on Core ML, https://developer.apple.com/videos/play/wwdc2017/506/.</p></div>
Claim 7	

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found³ (P.R. 3-1(b)-(c))
[7a.] A method as recited in claim 1, further comprising:	The '132 Patent Accused Products perform the method of claim 1. <i>See</i> [1a]-[1e].
[7b.] determining that the second portion of the digital image data does not represent a physical object of the predetermined type; and	The '132 Patent Accused Products perform determining that the second portion of the digital image data does not represent a physical object of the predetermined type. <i>See</i> [1b]-[1e].
[7c.] determining a different amount of gain to apply to the second portion than an amount of gain determined for the first portion.	The '132 Patent Accused Products perform determining a different amount of gain to apply to the second portion than an amount of gain determined for the first portion. <i>See</i> [1b]-[1e].
Claim 13	
[13a.] A method as recited in claim 7, wherein determining the amount of gain to apply to the first portion of the digital image data comprises:	The '132 Patent Accused Products perform the method of claim 7. <i>See</i> [7a]-[7c].
[13b.] modifying a gain map based on the determination that the first portion of the digital image data represents a physical object of the predetermined type; and	The '132 Patent Accused Products perform modifying a gain map based on the determination that the first portion of the digital image data represents a physical object of the predetermined type. <i>See</i> [1c]-[1e]. <i>See, e.g.:</i>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>White Vignette for Faces Filter Recipe</p> <p>Increases the brightness of an image at the periphery of a face detected in an image.</p> <p>Figure 5-2 The White Vignette filter processing chain</p>  <p>To create a white vignette filter:</p> <ul style="list-style-type: none"> • Find the human face in an image. • Create a base shade map using CIRadialGradient centered on the face. • Blend the base shade map with the original image. <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filer_recipes/ci_filter_recipes.html#apple_ref/doc/uid/TP30001185-CH4-SW22</p> <p>The autoAdjustmentFiltersWithOptions: method returns an array of options filters that you'll then want to chain together and apply to the analyzed image, as shown in Listing 3-1.</p>

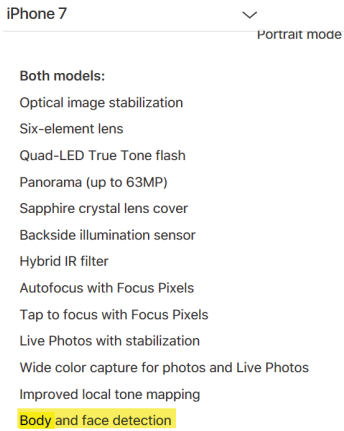
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))												
	<p>Table 3-1 Filters that Core Image uses to enhance an image</p> <table border="1"> <thead> <tr> <th>Filter</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>CIRedEyeCorrection</td><td>Repairs red/amber/white eye due to camera flash</td></tr> <tr> <td>CIFaceBalance</td><td>Adjusts the color of a face to give pleasing skin tones</td></tr> <tr> <td>CIVibrance</td><td>Increases the saturation of an image without distorting the skin tones</td></tr> <tr> <td>CIToneCurve</td><td>Adjusts image contrast</td></tr> <tr> <td>CIHighlightShadowAdjust</td><td>Adjusts shadow details</td></tr> </tbody> </table> <p>Core Image Programming Guide, Using Auto Enhancement Filters, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html.</p> <p>Create a Shade Map</p> <p>Use the CIRadialGradient filter to create a shade map centered on the face. The center of the shade map should be transparent so that the face in the image remains untouched. The edges of the map should be opaque white. Areas in between should have varying degrees of transparency.</p> <p>To achieve this effect, set the input parameters to CIRadialGradient as follows:</p> <ul style="list-style-type: none"> • Set inputRadius0 to a value larger than the longest dimension of the the image. • Set inputRadius1 to a value larger than the face, such as face.bounds.size.height + 50. • Set inputColor0 to opaque white. • Set inputColor1 to transparent white. • Set the inputCenter to the center of the face bounds that you computed with Listing 5-4. <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filer_recipes/ci_filer_recipes.html#apple_ref/doc/uid/TP30001185-CH4-SW22</p>	Filter	Purpose	CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash	CIFaceBalance	Adjusts the color of a face to give pleasing skin tones	CIVibrance	Increases the saturation of an image without distorting the skin tones	CIToneCurve	Adjusts image contrast	CIHighlightShadowAdjust	Adjusts shadow details
Filter	Purpose												
CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash												
CIFaceBalance	Adjusts the color of a face to give pleasing skin tones												
CIVibrance	Increases the saturation of an image without distorting the skin tones												
CIToneCurve	Adjusts image contrast												
CIHighlightShadowAdjust	Adjusts shadow details												


Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found³ (P.R. 3-1(b)-(c))
[13c.] wherein applying the determined amount of gain to apply to the first portion of the digital image data comprises obtaining the amount of gain to apply to the first portion of the digital image data from the gain map.	The '132 Patent Accused Products perform obtaining the amount of gain to apply to the first portion of the digital image data from the gain map. See [13b].
Claim 21	
[21a.] An image processing unit comprising:	To the extent the preamble is found to be limiting, the '132 Patent Accused Products include an image processing unit. See [1a].
[21b.] an object detection unit to receive an intensity value for each of a plurality of pixels in a set of digital image data and to determine that a first portion of the set of digital image data represents a physical object of a predetermined type; and	The '132 Patent Accused Products include an object detection unit to receive an intensity value for each of a plurality of pixels in a set of digital image data and to determine that a first portion of the set of digital image data represents a physical object of a predetermined type. For example, the '132 Patent Accused Products receive digital image data from the camera that includes an intensity value for each of a plurality of pixels. See, e.g., https://www.apple.com/iphone-7/specs/ (identifying a 12 Megapixel camera in the iPhone 7 and 12 Megapixel wide-angle and telephoto cameras in the iPhone 7 Plus); https://www.apple.com/iphone-8/specs/ identifying a 12 Megapixel camera in the iPhone 8 and 12 Megapixel wide-angle and telephoto cameras in the iPhone 8 Plus); https://www.apple.com/iphone-x/specs/ (identifying 12 Megapixel wide-angle and telephoto cameras in the iPhone X). The iPhone 5, 5s, 5c, 6, 6 Plus, 6s, SE sold after May 21, 2013 and running iOS 5.0+ or later also include a camera, which provide an intensity value for each of a plurality of pixels in a set of digital image data to an object detection unit. See, e.g., http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-5-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-5s-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-5c-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-6-and-iphone-6-plus-teardown/ ; http://www.techinsights.com/about-techinsights/overview/blog/inside-the-iphone-6s/ ; http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-se-teardown/ . The iPad (3 rd generation), iPad (4 th generation), iPad Air, iPad Air 2, iPad (2017), iPad Mini, iPad Mini 2, iPad Mini 3, iPad Mini 4, iPad Pro (1 st generation), and iPad Pro (2 nd generation) each also include a camera. See, e.g.,

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p> https://support.apple.com/kb/SP647?locale=en_US; https://support.apple.com/kb/SP662?locale=en_US; https://support.apple.com/kb/SP692?locale=en_US; https://support.apple.com/kb/SP708?viewlocale=en_US&locale=en_US; https://www.apple.com/ipad-9.7/specs/; https://support.apple.com/kb/SP661?locale=en_US; https://support.apple.com/kb/SP693?locale=en_US; https://support.apple.com/kb/SP709?locale=en_US&viewlocale=en_US; https://www.apple.com/ipad-mini-4/specs/. </p> <p>The intensity values may also be provided to the object detection unit by memory in each of the '132 Patent Accused Products.</p> <p>The digital image data may be received by a CPU, a GPU, and/or an ISP of the '132 Patent Accused Products. <i>See</i> [1a].</p> <p>The '132 Patent Accused Products can detect objects such as bodies and faces of subjects within a picture. <i>See, e.g.,</i>:</p> <div data-bbox="505 980 846 1407">  <p>The screenshot shows the iPhone 7 camera settings menu. At the top, 'iPhone 7' is selected, and 'Portrait mode' is shown with a dropdown arrow. Below this, under the heading 'Both models:', a list of features is displayed: Optical image stabilization, Six-element lens, Quad-LED True Tone flash, Panorama (up to 63MP), Sapphire crystal lens cover, Backside illumination sensor, Hybrid IR filter, Autofocus with Focus Pixels, Tap to focus with Focus Pixels, Live Photos with stabilization, Wide color capture for photos and Live Photos, Improved local tone mapping, and 'Body and face detection' which is highlighted with a yellow background.</p> </div>

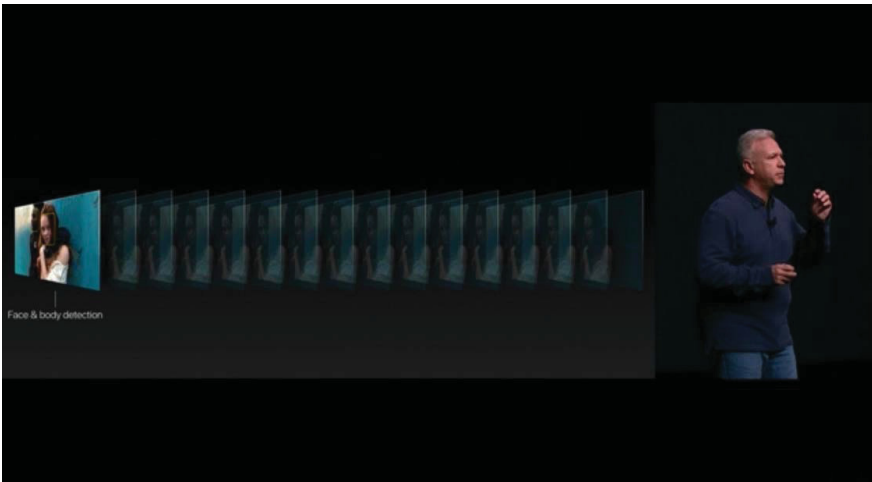
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

<p>'132 Patent Claim² (P.R. 3-1(a))</p>	<p>The '132 Patent Accused Products And Where Each Claim Element Is Found³ (P.R. 3-1(b)-(c))</p>
	<p>https://www.apple.com/iphone-7/specs/ (identifying “body and face detection” as a feature).</p> <div data-bbox="511 779 850 1199"> <p>iPhone 8 </p> <hr/> <p>Both models:</p> <ul style="list-style-type: none"> Optical image stabilization Six-element lens Quad-LED True Tone flash with Slow Sync Panorama (up to 63MP) Sapphire crystal lens cover Backside illumination sensor Hybrid IR filter Autofocus with Focus Pixels Tap to focus with Focus Pixels Live Photos with stabilization Wide color capture for photos and Live Photos Improved local tone mapping Body and face detection </div> <p>https://www.apple.com/iphone-8/specs/ (identifying “body and face detection” as a feature of the iPhone 8 and the iPhone 8 Plus).</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>iPhone X <small>aperture: f/1.8 aperture</small> ▼</p> <p>Telephoto: f/2.4 aperture</p> <p>Optical zoom; digital zoom up to 10x</p> <p>Portrait mode</p> <p>Portrait Lighting (beta)</p> <p>Dual optical image stabilization</p> <p>Six-element lens</p> <p>Quad-LED True Tone flash with Slow Sync</p> <p>Panorama (up to 63MP)</p> <p>Sapphire crystal lens cover</p> <p>Backside illumination sensor</p> <p>Hybrid IR filter</p> <p>Autofocus with Focus Pixels</p> <p>Tap to focus with Focus Pixels</p> <p>Live Photos with stabilization</p> <p>Wide color capture for photos and Live Photos</p> <p>Improved local tone mapping</p> <p>Body and face detection</p> <p>https://www.apple.com/iphone-x/specs/ (identifying “body and face detection” as a feature of the iPhone X).</p> <p>Apple provides a framework, instruction for using that framework, and software that uses that framework. <i>See, e.g.,</i>:</p> <p style="padding-left: 40px;">Core Image is an image processing and analysis technology that provides high-performance processing for still and video images.</p> <p>https://developer.apple.com/documentation/coreimage.</p> <p style="padding-left: 40px;">Core Image can analyze and find human faces in an image. . . . After Core Image detects a face, it can provide information about</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>face features, such as eye and mouth positions. It can also track the position an identified face in a video.</p> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html#apple_ref/doc/uid/TP30001185-CH8-SW1</p>  <p><u>What does this ISP (Image Signal Processor) do?</u> Well, I'm going to take a quick walk through some of the things the ISP does every time we take a picture. It's unbelievable. Well, the first thing it does is it reads the scene and uses machine learning to look for objects and people and bodies within it.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Apple Special Event, September 7, 2016 at 1:03:45, https://www.apple.com/lae/apple-events/september-2016/.</p>  <p>Last year we introduced Portrait mode. iPhone 8 takes fantastic Portrait modes and people flip over taking these photos and now with the iPhone 8 Plus, you're going to get sharper details, more uses in low light and even a more natural bokeh in the background of the photos.</p> <p>Apple Special Event, September 12, 2017 at 1:02:35, https://www.apple.com/apple-events/september-2017/.</p> <p>And iPhone 10 is fantastic for the Portrait mode feature that we all love. iPhone 10, you get great portrait mode especially in</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>lower light.</p> <p>Apple Special Event, September 12, 2017 at 1:39:40, https://www.apple.com/apple-events/september-2017/.</p> <p>Choose a photo mode. Camera has several photo modes so you can shoot standard and square-format photos, and panoramas. To choose a mode, swipe the screen left or right, or tap Photo, Square, or Pano.</p> <p>On iPhone X, iPhone 8 Plus, and iPhone 7 Plus, you can also choose Portrait mode, which uses a depth effect to focus on your subject and blur the background.</p> <p>iPhone User Guide For iOS 11.2 software at Chapter "Camera," available at https://help.apple.com/iphone/11/. See also https://support.apple.com/en-us/HT208118 ("[Portrait mode] lets you compose a photo that keeps your subject sharp while creating a beautifully blurred background.").</p> <p>Core Image can detect human face features in still images and track them over time in video images. Knowing where faces are can help you determine where to place a vignette or apply other special filters.</p> <p>Core Image Programming Guide, About Core Image, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_intro/ci_intro.html#//apple_ref/doc/uid/TP30001185.</p> <p>Core Image can analyze and find human faces in an image. It performs face detection, not recognition. Face detection is the identification of rectangles that contain human face features, whereas face recognition is the identification of specific human faces (John, Mary, and so on). After Core Image detects a face, it can provide information about face features, such as eye and mouth positions. It can also track the position an identified face in a video.</p> <p>Knowing where the faces are in an image lets you perform other operations, such as cropping or adjusting the image quality of the face (tone balance, red-eye correction and so on). You can also perform other interesting operations on the faces; for example:</p> <p>Anonymous Faces Filter Recipe shows how to apply a pixellate filter only to the faces in an image.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>White Vignette for Faces Filter Recipe shows how to place a vignette around a face.</p> <p>Core Image Programming Guide, Detecting Faces in Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html#apple_ref/doc/uid/TP30001185-CH8-SW1.</p> <p>Model barcode data to create barcode images with Core Image filters or to interpret barcodes detected by Core Image, Vision, or AVFoundation features.</p> <p>class CIBarcodeDescriptor An abstract base class that represents a machine readable code's attributes.</p> <p>class CIQRCodeDescriptor A concrete subclass of CIBarcodeDescriptor that represents a square QR code symbol.</p> <p>class CIAztecCodeDescriptor A concrete subclass of CIBarcodeDescriptor that represents an Aztec code symbol.</p> <p>class CIPDF417CodeDescriptor A concrete subclass of CIBarcodeDescriptor that represents a PDF 417 symbol.</p> <p>class CIDataMatrixCodeDescriptor A concrete subclass of CIBarcodeDescriptor that represents a Data Matrix code symbol.</p> <p>Core Image Framework Documentation, https://developer.apple.com/documentation/coreimage.</p> <p>class CIDetector</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>An image processor that identifies notable features (such as faces and barcodes) in a still image or video.</p> <p>class CIFeature</p> <p>The abstract superclass for objects representing notable features detected in an image.</p> <p>class CIFaceFeature</p> <p>Information about a face detected in a still or video image.</p> <p>class CIRectangleFeature</p> <p>Information about a rectangular region detected in a still or video image.</p> <p>class CITextFeature</p> <p>Information about a region likely to contain text detected in a still or video image.</p> <p>class CIQRCodeFeature</p> <p>Information about a Quick Response code (a kind of 2D barcode) detected in a still or video image.</p> <p>Core Image Framework Documentation, https://developer.apple.com/documentation/coreimage.</p> <p>Overview</p> <p>A CIDetector object uses image processing to search for and identify notable features (faces, rectangles, and barcodes) in a still image or video. Detected features are represented by CIFeature objects that provide more information about each feature.</p> <p>Core Image Framework Documentation, https://developer.apple.com/documentation/coreimage/cidetector.</p> <p>let CIDetectorTypeFace: String</p> <p>A detector that searches for faces in a still image or video, returning CIFaceFeature objects that provide information about detected faces.</p> <p>let CIDetectorTypeRectangle: String</p> <p>A detector that searches for rectangular areas in a still image or video, returning CIRectangleFeature objects that provide information about detected regions.</p> <p>let CIDetectorTypeQRCode: String</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>A detector that searches for Quick Response codes (a type of 2D barcode) in a still image or video, returning CIQRCodeFeature objects that provide information about detected barcodes. let CIDetectorTypeText: String A detector that searches for text in a still image or video, returning CITextFeature objects that provide information about detected regions.</p> <p>Core Image Framework Documentation, Detector Types, https://developer.apple.com/documentation/coreimage/cidetector/detector_types.</p> <p>CIDetectorTypeFace A detector that searches for faces in a still image or video, returning CIFaceFeature objects that provide information about detected faces.</p> <p>Core Image Framework Documentation, CIDetectorTypeFace, https://developer.apple.com/documentation/coreimage/cidetector/typeface.</p> <p>CIFaceFeature Information about a face detected in a still or video image.</p> <p>Overview The properties of a CIFaceFeature object provide information about the face's eyes and mouth. A face object in a video can also have properties that track its location over time—tracking ID and frame count.</p> <p>Core Image Framework Documentation, CIFaceFeature, https://developer.apple.com/documentation/coreimage/cifacefeature.</p> <p>let CIDetectorEyeBlink: String An option for whether Core Image will perform additional processing to recognize closed eyes in detected faces. let CIDetectorSmile: String An option for whether Core Image will perform additional processing to recognize smiles in detected faces.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Core Image Framework Documentation, Feature Detection Keys, https://developer.apple.com/documentation/coreimage/cidetector/feature_detection_keys.</p> <p>Vision Apply high-performance image analysis and computer vision techniques to identify faces, detect features, and classify scenes in images and video.</p> <p>Vision Framework for iOS 11.0+, https://developer.apple.com/documentation/vision.</p> <p>Face Detection and Recognition class VNDetectFaceRectanglesRequest An image analysis request that finds faces within an image. class VNDetectFaceLandmarksRequest An image analysis request that finds facial features (such as the eyes and mouth) in an image. class VNFaceObservation Face or facial-feature information detected by an image analysis request.</p> <p>Vision Framework for iOS 11.0+, https://developer.apple.com/documentation/vision.</p> <p>var landmarks: VNFaceLandmarks2D? The facial features found in the detected face. class VNFaceLandmarks2D A collection of all facial features detected in a face observation, with 2D geometry information for each. class VNFaceLandmarkRegion2D 2D geometry information for a specific facial feature. class VNFaceLandmarks The abstract superclass for containers of face landmark information. class VNFaceLandmarkRegion The abstract superclass for information about a specific face landmark.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Vision Framework for iOS 11.0+, VNFaceObservation, https://developer.apple.com/documentation/vision/vnfaceobservation.</p> <p>class VNDetectTextRectanglesRequest An image analysis request that finds regions of visible text in an image. class VNTextObservation Information about regions of text detected by an image analysis request.</p> <p>Vision Framework for iOS 11.0+, https://developer.apple.com/documentation/vision.</p> <p>class VNDetectRectanglesRequest An image analysis request that finds projected rectangular regions in an image.</p> <p>Vision Framework for iOS 11.0+, https://developer.apple.com/documentation/vision.</p> <p>We implement several strategies to minimize memory footprint and GPU usage. To reduce memory footprint, we allocate the intermediate layers of our neural networks by analyzing the compute graph. This allows us to alias multiple layers to the same buffer. While being fully deterministic, this technique reduces memory footprint without impacting the performance or allocations fragmentation, and can be used on either the CPU or GPU.</p> <p>An On-Device Deep Neural Network for Face Detection, https://machinelearning.apple.com/2017/11/16/face-detection.html.</p> <p>Now I ask myRequestHandler to perform my request and this in this case [inaudible] it's just one request I have my array, but it could be many and I get my observations back.</p> <p>And this can be many faces that I detected. Now the one thing I would like to highlight here is the results come back as part of the request that we actually set up initially. How does it look when we want to track something? We create a sequence request handler [inaudible] of course not set up as an image because we have to [inaudible] all the frames of the sequence. So, I started with an observation that I got from the previous detection or I mark something up and I create my tracking request. And I simply have to run the request.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>And I feed in in this case as a pixel buffer the frame that is currently being dragged.</p> <p>And out of it again I get some results. So, now that we have talked about how this API is kind of structured I would like to guide you through some best practices so that you get, you know, the best experience out of Vision. So, when we want to put together a Computer Vision task you have to think about a few things. Number one, what is the right image type that I want to use. Number two, what am I going to do with the image.</p> <p>And number three, what performance do I need or want. Of course, you always want fastest, but there are some tradeoffs that you have to think about.</p> <p>So, let's talk about the image type. Vision supports a number of image types and they range from CVPixelBuffer, CGImage or even as we saw in the previous example just from data that I use in NSURL. And we go over all these types in the following slides so that you know what to choose when. Which to choose depends a lot of like what you want to do. If you run from a camera stream or if you run from files on disk you have to look at that [inaudible] kind of which type of image you want to use.</p> <p>Now two important things to remember is we already have an imaging pipeline in the Vision framework you don't need to scale the images. So, unless you already have a very small representation that you absolutely want to use, please don't pre-scale because we'll just do the work twice. And mind the orientation. Computer Vision algorithms are mostly not, you know, sensitive to orientation or sorry, they are sensitive to orientation so you have to pass that in. And that is an important part because if you pass in a portrait image that's actually lying on its side we will not find the faces and that's one of the common mistakes that usually happens. So, I promised to go over the types. When you want to do something streaming we want to use the CVPixelBuffer. When you create a VideoDataOut [inaudible] capture you will get CMSampleBuffers and through those we get your CVPixelBuffers.</p> <p>It's also a pretty good format if you already have something where you keep your image data raw in memory like it's LGB pixels and wrap them into a CVPixelBuffer this is a great format to pass into Vision. When you get files from disk please use the URL or if it comes from the web use the NSData path. The great thing about that is it really allows us to reduce the memory for print in your application.</p> <p>Vision will only read what it needs to perform the task. If you think about you want to do face detection on a 64-megapixel</p>

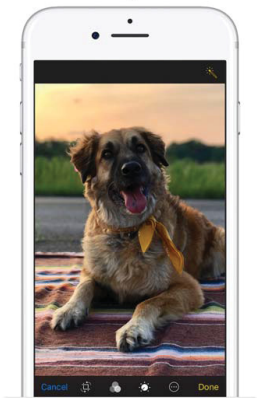
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>panorama Vision will actually reduce your memory for it, but not reading the full file actually into the memory and that is an important thing to keep in mind. We will read in this case the EXIF Orientation out of the file, but you can override it if you have to for those formats that don't support it.</p> <p>If you're already using Core Image in your application by all means process the CI image.</p> <p>This is also important when you want to actually do some preprocessing.</p> <p>If you have some domain knowledge of what you want to do in your Computer Vision task you can do some preprocessing and try and enhance the image and, therefore, enhance the Vision results. If you want to learn a bit more about Core Image, there's a session on Thursday at 1:50 and they will also show the integration with our Vision framework. Last but not least, if you have all the images in your UI you can use the CG image [inaudible] out of the NS image or the UI images let's say it comes from the UI image picker and pass those into Vision. Now what am I going to do with the image and that's where we have to decide if I want to do something interactive with the image in that case I use my ImageRequestHandler. It will hold on to the image for the time and I can do multiple passes on that image and get the best results out of that.</p> <p>WWDC 2017 - Session 506 - iOS, macOS, tvOS, Vision Framework: Building on Core ML, https://developer.apple.com/videos/play/wwdc2017/506/</p> <p>But keep in mind that this completion handler gets called on that queue in which you actually set it off. So, if you need to update your UI you have to dispatch that back to the main queue. So as Brett already highlighted, we have a new face detection and you might say oh God, yet another one. But we have good reasons for this. Vision uses deep learning and this gives us really a lot better precision and recall, therefore, much better results. The downside of it, on older hardware it will run a bit slower. So, let's look a little bit at our overall landscape of face detectors that we have available.</p> <p>So, we have Vision which really gives us the best results and it's pretty fast and also pretty good in its power use as it is optimized for that. And we have it available on all platforms except the watchOS. And this is the same in terms of availability for Core Image and it's a bit faster, but the results are not quite as good. In the AV capture session which is only happening during the capture side we can actually use hardware so it's really fast in performance, but the results again are not as good as we get out of Vision.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>So, you have to choose depending on your application what you want to do choose the right technology for the face detection. Now I did mention that our quality is better, so let me try to prove that a little bit. So, I have here an image and I ran the face detection through Core Image. And we find two faces and we see roughly where the eyes and where the mouth are. Now in Vision we find all four faces even the occluded ones and we get a whole lot more details with the visual landmarks.</p> <p>Speaking of Core Image, I would like to highlight a little bit what's happening with the CIDetectors. So, whoever uses it already can keep on using them they are still in Core Image, but all new parts and all the improvements in terms of algorithms for computer moving forward will be in Vision that's the new home for Computer Vision.</p> <p>WWDC 2017 - Session 506 - iOS, macOS, tvOS, Vision Framework: Building on Core ML, https://developer.apple.com/videos/play/wwdc2017/506/.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Enhance instantly</p>  <p>Auto-enhance will instantly improve your picture's exposure, contrast, saturation, and other qualities. It's the fastest way to make your photo even more amazing.</p> <p>https://support.apple.com/en-us/ht205659</p> <p>Additionally, the iPhone 7 Plus, iPhone 8 Plus, and iPhone X support Portrait mode for detecting faces and applying corrections:</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div data-bbox="511 745 779 1281"> </div> <div data-bbox="803 766 1161 1081"> <p>Add Portrait Lighting</p> <p>On iPhone 8 Plus and iPhone X, you can use the Portrait Lighting feature to apply four impressive studio-quality lighting effects to your Portrait mode images. Choose from Studio Light to brighten facial features, Contour Light for more dramatic directional lighting, Stage Light to isolate your subject in the spotlight, or Stage Mono for stage light in a classic black and white.</p> <ol style="list-style-type: none"> 1. Select the photo that you want to change. 2. Tap Edit. 3. Swipe to choose your Lighting effect. 4. Tap Done. <p>Portrait Lighting is currently in beta.</p> </div> <p data-bbox="446 1312 1515 1360">https://support.apple.com/en-us/HT208118; see also https://www.apple.com/newsroom/2016/10/portrait-mode-now-available-on-iphone-7-plus-with-ios-101/</p>
[21c.] a correction unit to determine a	The 132 Patent Accused Products include a correction unit to determine a type or amount of correction to apply to the first portion of the

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))												
type or amount of correction to apply to the first portion of the set of digital image data, based on an output of the object detection unit, and to apply the determined type or amount of correction to the first portion of the set of digital image data, and	<p>set of digital image data, based on an output of the object detection unit, and to apply the determined type or amount of correction to the first portion of the set of digital image data.</p> <p>The corrections may be determined and applied by a CPU, a GPU, and/or an ISP of the '132 Patent Accused Products. <i>See</i> [1a].</p> <p><i>See, e.g.,</i>:</p> <p>Core Image is an image processing and analysis technology that provides high-performance processing for still and video images.</p> <p>https://developer.apple.com/documentation/coreimage.</p> <p>Table 3-1 Filters that Core Image uses to enhance an image</p> <table border="1"> <thead> <tr> <th>Filter</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>CIRedEyeCorrection</td><td>Repairs red/amber/white eye due to camera flash</td></tr> <tr> <td>CIFaceBalance</td><td>Adjusts the color of a face to give pleasing skin tones</td></tr> <tr> <td>CIVibrance</td><td>Increases the saturation of an image without distorting the skin tones</td></tr> <tr> <td>CIToneCurve</td><td>Adjusts image contrast</td></tr> <tr> <td>CIHighlightShadowAdjust</td><td>Adjusts shadow details</td></tr> </tbody> </table> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html#//apple_ref/doc/uid/TP30001185-CH11-SW1</p>	Filter	Purpose	CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash	CIFaceBalance	Adjusts the color of a face to give pleasing skin tones	CIVibrance	Increases the saturation of an image without distorting the skin tones	CIToneCurve	Adjusts image contrast	CIHighlightShadowAdjust	Adjusts shadow details
Filter	Purpose												
CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash												
CIFaceBalance	Adjusts the color of a face to give pleasing skin tones												
CIVibrance	Increases the saturation of an image without distorting the skin tones												
CIToneCurve	Adjusts image contrast												
CIHighlightShadowAdjust	Adjusts shadow details												

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>As another example, the iPhone 7Plus, iPhone 8Plus, and iPhone X each can operate in Portrait mode. When using Portrait mode, the iPhone 7Plus, iPhone 8Plus, and iPhone X can apply a correction to the first portion of the set of digital image data, <i>i.e.</i> the subjects of the picture.</p> <p style="text-align: center;">Choose a photo mode. Camera has several photo modes so you can shoot standard and square-format photos, and panoramas. To choose a mode, swipe the screen left or right, or tap Photo, Square, or Pano.</p> <p style="text-align: center;">On iPhone X, iPhone 8 Plus, and iPhone 7 Plus, you can also choose Portrait mode, which uses a depth effect to focus on your subject and blur the background.</p> <p>iPhone User Guide For iOS 11.2 software at Chapter “Camera,” <i>available at</i> https://help.apple.com/iphone/11/.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div data-bbox="867 737 1097 1199" data-label="Image"> </div> <p data-bbox="508 1220 1433 1245">[Portrait mode] lets you compose a photo that keeps your subject sharp while creating a beautifully blurred background.</p> <p data-bbox="449 1264 784 1289">https://support.apple.com/en-us/HT208118.</p> <p data-bbox="508 1308 1433 1354"><i>See also</i> iPhone User Guide For iOS 11.2 software at Chapter “Camera,” available at https://help.apple.com/iphone/11/ (discussing different lighting effects that can be applied to the subject of a picture shot in Portrait mode).</p> <p data-bbox="508 1373 1487 1419">Core Image can detect human face features in still images and track them over time in video images. Knowing where faces are can help you determine where to place a vignette or apply other special filters.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Core Image Programming Guide, About Core Image, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_intro/ci_intro.html#apple_ref/doc/uid/TP30001185.</p> <p>Core Image can analyze and find human faces in an image. It performs face detection, not recognition. Face detection is the identification of rectangles that contain human face features, whereas face recognition is the identification of specific human faces (John, Mary, and so on). After Core Image detects a face, it can provide information about face features, such as eye and mouth positions. It can also track the position an identified face in a video.</p> <p>Knowing where the faces are in an image lets you perform other operations, such as cropping or adjusting the image quality of the face (tone balance, red-eye correction and so on). You can also perform other interesting operations on the faces; for example:</p> <p>Anonymous Faces Filter Recipe shows how to apply a pixellate filter only to the faces in an image.</p> <p>White Vignette for Faces Filter Recipe shows how to place a vignette around a face.</p> <p>Core Image Programming Guide, Detecting Faces in Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html#apple_ref/doc/uid/TP30001185-CH8-SW1.</p> <p>The auto enhancement feature of Core Image analyzes an image for its histogram, face region contents, and metadata properties. It then returns an array of CIFilter objects whose input parameters are already set to values that will improve the analyzed image. Auto enhancement is available in iOS v5.0 and later and in OS X v10.8 and later.</p> <p>Core Image Programming Guide, Auto Enhancing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))												
	<p>The autoAdjustmentFiltersWithOptions: method returns an array of options filters that you'll then want to chain together and apply to the analyzed image, as shown in Listing 3-1.</p> <p>Table 3-1 Filters that Core Image uses to enhance an image</p> <table border="1"> <thead> <tr> <th>Filter</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>CIRedEyeCorrection</td><td>Repairs red/amber/white eye due to camera flash</td></tr> <tr> <td>CIFaceBalance</td><td>Adjusts the color of a face to give pleasing skin tones</td></tr> <tr> <td>CIVibrance</td><td>Increases the saturation of an image without distorting the skin tones</td></tr> <tr> <td>CIToneCurve</td><td>Adjusts image contrast</td></tr> <tr> <td>CIHighlightShadowAdjust</td><td>Adjusts shadow details</td></tr> </tbody> </table> <p>Core Image Programming Guide, Using Auto Enhancement Filters, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html.</p> <p>Return Value An array of CIFilter instances preconfigured for correcting deficiencies in the supplied image.</p> <p>Apple Developer Documentation, autoAdjustFilters(options:), https://developer.apple.com/documentation/coreimage/ciimage/1437792-autoadjustmentfilters.</p>	Filter	Purpose	CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash	CIFaceBalance	Adjusts the color of a face to give pleasing skin tones	CIVibrance	Increases the saturation of an image without distorting the skin tones	CIToneCurve	Adjusts image contrast	CIHighlightShadowAdjust	Adjusts shadow details
Filter	Purpose												
CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash												
CIFaceBalance	Adjusts the color of a face to give pleasing skin tones												
CIVibrance	Increases the saturation of an image without distorting the skin tones												
CIToneCurve	Adjusts image contrast												
CIHighlightShadowAdjust	Adjusts shadow details												

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))																												
	<p>Table 4-1 Filter category constants for effect types</p> <table border="1"> <thead> <tr> <th>Effect type</th><th>Indicates</th></tr> </thead> <tbody> <tr> <td>kCICategoryDistortionEffect</td><td>Distortion effects, such as bump, twirl, hole</td></tr> <tr> <td>kCICategoryGeometryAdjustment</td><td>Geometry adjustment, such as affine transform, crop, perspective transform</td></tr> <tr> <td>kCICategoryCompositeOperation</td><td>Compositing, such as source over, minimum, source atop, color dodge blend mode</td></tr> <tr> <td>kCICategoryHalftoneEffect</td><td>Halftone effects, such as screen, line screen, hatched</td></tr> <tr> <td>kCICategoryColorAdjustment</td><td>Color adjustment, such as gamma adjust, white point adjust, exposure</td></tr> <tr> <td>kCICategoryColorEffect</td><td>Color effect, such as hue adjust, posterize</td></tr> <tr> <td>kCICategoryTransition</td><td>Transitions between images, such as dissolve, disintegrate with mask, swipe</td></tr> <tr> <td>kCICategoryTileEffect</td><td>Tile effect, such as parallelogram, triangle</td></tr> <tr> <td>kCICategoryGenerator</td><td>Image generator, such as stripes, constant color, checkerboard</td></tr> <tr> <td>kCICategoryGradient</td><td>Gradient, such as axial, radial, Gaussian</td></tr> <tr> <td>kCICategoryStylize</td><td>Stylize, such as pixelate, crystallize</td></tr> <tr> <td>kCICategorySharpen</td><td>Sharpen, luminance</td></tr> <tr> <td>kCICategoryBlur</td><td>Blur, such as Gaussian, zoom, motion</td></tr> </tbody> </table> <p>Core Image Programming Guide, Querying the System for Filters, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_concepts/ci_concepts.html.</p> <p>Creating Custom Filters If the filters provided by Core Image don't provide the functionality you need, you can write your own filter. You can include a filter as part of an application project, or (in macOS only) you can package one or more filters as a standalone image unit. Image units use the NSBundle class and allow an app to host external plug-in filters. The following sections provide detailed information on how to create and use custom filters and image units: Expressing Image Processing Operations in Core Image Creating a Custom Filter describes the methods that you need to implement and other filter requirements. Using Your Own Custom Filter tells what's need for you to use the filter in your own app. (If you want to package it as a standalone image unit, see Packaging and Loading Image Units.)</p>	Effect type	Indicates	kCICategoryDistortionEffect	Distortion effects, such as bump, twirl, hole	kCICategoryGeometryAdjustment	Geometry adjustment, such as affine transform, crop, perspective transform	kCICategoryCompositeOperation	Compositing, such as source over, minimum, source atop, color dodge blend mode	kCICategoryHalftoneEffect	Halftone effects, such as screen, line screen, hatched	kCICategoryColorAdjustment	Color adjustment, such as gamma adjust, white point adjust, exposure	kCICategoryColorEffect	Color effect, such as hue adjust, posterize	kCICategoryTransition	Transitions between images, such as dissolve, disintegrate with mask, swipe	kCICategoryTileEffect	Tile effect, such as parallelogram, triangle	kCICategoryGenerator	Image generator, such as stripes, constant color, checkerboard	kCICategoryGradient	Gradient, such as axial, radial, Gaussian	kCICategoryStylize	Stylize, such as pixelate, crystallize	kCICategorySharpen	Sharpen, luminance	kCICategoryBlur	Blur, such as Gaussian, zoom, motion
Effect type	Indicates																												
kCICategoryDistortionEffect	Distortion effects, such as bump, twirl, hole																												
kCICategoryGeometryAdjustment	Geometry adjustment, such as affine transform, crop, perspective transform																												
kCICategoryCompositeOperation	Compositing, such as source over, minimum, source atop, color dodge blend mode																												
kCICategoryHalftoneEffect	Halftone effects, such as screen, line screen, hatched																												
kCICategoryColorAdjustment	Color adjustment, such as gamma adjust, white point adjust, exposure																												
kCICategoryColorEffect	Color effect, such as hue adjust, posterize																												
kCICategoryTransition	Transitions between images, such as dissolve, disintegrate with mask, swipe																												
kCICategoryTileEffect	Tile effect, such as parallelogram, triangle																												
kCICategoryGenerator	Image generator, such as stripes, constant color, checkerboard																												
kCICategoryGradient	Gradient, such as axial, radial, Gaussian																												
kCICategoryStylize	Stylize, such as pixelate, crystallize																												
kCICategorySharpen	Sharpen, luminance																												
kCICategoryBlur	Blur, such as Gaussian, zoom, motion																												

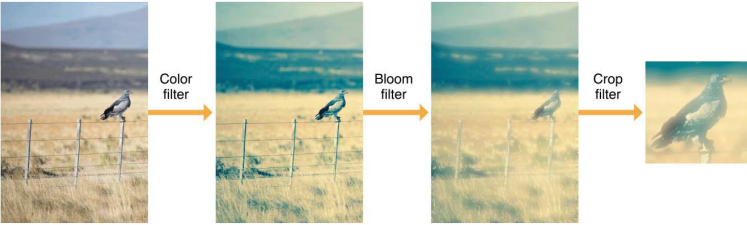
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Supplying an ROI Function provides information about the region of interest and when you must supply a method to calculate this region. (It's not always needed.)</p> <p>Writing Nonexecutable Filters is a must-read section for anyone who plans to write a filter that is CPU nonexecutable, as it lists the requirements for such filters. An image unit can contain both kinds of filters. CPU nonexecutable filters are secure because they cannot harbor viruses and Trojan horses. Filter clients who are security conscious may want to use only those filters that are CPU nonexecutable.</p> <p>Kernel Routine Examples provides kernel routines for three sample filters: brightening, multiply, and hole distortion.</p> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_custom_filters/ci_custom_filters.html#/apple_ref/doc/uid/TP30001185-CH6-TPXREF101</p> <p>Creating a Custom Filter</p> <p>This section shows how to create a Core Image filter that has an Objective-C portion and a kernel portion. By following the steps in this section, you'll create a filter that is CPU executable. You can package this filter, along with other filters if you'd like, as an image unit by following the instructions in Packaging and Loading Image Units. Or, you can simply use the filter from within your own app. See Using Your Own Custom Filter for details.</p> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_custom_filters/ci_custom_filters.html#/apple_ref/doc/uid/TP30001185-CH6-TPXREF101</p> <p>Packaging and Loading Image Units</p> <p>An image unit represents the plug-in architecture for Core Image filters. Image units use the NSBundle class as the packaging mechanism to allow you to make the filters that you create available to other apps. An image unit can contain filters that are executable or nonexecutable. (See Executable and Nonexecutable Filters for details.)</p> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_image_units/ci_image_units.html#/apple_ref/doc/uid/TP30001185-CH7-SW12</p> <p>Overview</p>

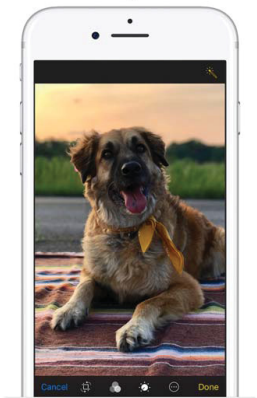
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>The CIFilterGenerator class provides methods for creating a CIFilter object by chaining together existing CIFilter objects to create complex effects. (A filter chain refers to the CIFilter objects that are connected in the CIFilterGenerator object.) The complex effect can be encapsulated as a CIFilterGenerator object and saved as a file so that it can be used again. The filter generator file contains an archived instance of all the CIFilter objects that are chained together.</p> <p>Any filter generator files that you copy to /Library/Graphics/Image Units/ are loaded when any of the loading methods provided by the CIPugin class are invoked. A CIFilterGenerator object is registered by its filename or, if present, by a class attribute that you supply in its description.</p> <p>You can create a CIFilterGenerator object programmatically, using the methods provided by the CIFilterGenerator class, or by using the editor view provided by Core Image.</p> <p>Core Image Documentation, CIFilterGenerator, https://developer.apple.com/documentation/coreimage/cifiltergenerator.</p> <p>sourceKey The key that specifies the source object. For example, if the source is the output image of a filter, pass the outputImage key. Pass nil if the source object is used directly.</p> <p>Core Image Documentation, connect(_ :withKey:to:withKey:), https://developer.apple.com/documentation/coreimage/cifiltergenerator/1438159-connect.</p> <p>Core Image filters process and produce Core Image images. A CIImage instance is an immutable object representing an image. These objects don't directly represent image bitmap data—instead, a CIImage object is a “recipe” for producing an image. One recipe might call for loading an image from a file; another might represent output from a filter, or from a chain of filters. Core Image performs these recipes only when you request that an image be rendered for display or output.</p> <p>Core Image Programming Guide, Processing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_tasks/ci_tasks.html.</p> <p>Chaining Filters for Complex Effects Every Core Image filter produces an output CIImage object, so you can use this object as input to another filter. For example, the</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>sequence of filters illustrated in Figure 1-1 applies a color effect to an image, then adds a glow effect, and finally crops a section out of the result.</p> <p>Figure 1-1 Construct a filter chain by connecting filter inputs and outputs</p>  <p>Core Image Programming Guide, Processing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_tasks/ci_tasks.html</p> <p>Additionally, correction may be determined and applied by extensions. <i>See, e.g.,</i>:</p> <p>Third-party apps and extensions expand your editing options in Photos for iOS and macOS. You can apply edits from multiple apps and extensions to one photo, or use any combination of apps and extensions plus the editing tools built into Photos.</p> <p>https://support.apple.com/en-us/HT205245</p>

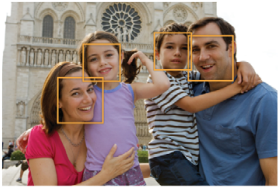
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Enhance instantly</p>  <p>Auto-enhance will instantly improve your picture's exposure, contrast, saturation, and other qualities. It's the fastest way to make your photo even more amazing.</p> <p>https://support.apple.com/en-us/ht205659</p> <p>Additionally, the iPhone 7 Plus, iPhone 8 Plus, and iPhone X support Portrait mode for detecting faces and applying corrections:</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div data-bbox="511 745 779 1281"> </div> <div data-bbox="803 766 1161 1081"> <p>Add Portrait Lighting</p> <p>On iPhone 8 Plus and iPhone X, you can use the Portrait Lighting feature to apply four impressive studio-quality lighting effects to your Portrait mode images. Choose from Studio Light to brighten facial features, Contour Light for more dramatic directional lighting, Stage Light to isolate your subject in the spotlight, or Stage Mono for stage light in a classic black and white.</p> <ol style="list-style-type: none"> 1. Select the photo that you want to change. 2. Tap Edit. 3. Swipe to choose your Lighting effect. 4. Tap Done. <p>Portrait Lighting is currently in beta.</p> </div> <p data-bbox="446 1312 1518 1354">https://support.apple.com/en-us/HT208118; see also https://www.apple.com/newsroom/2016/10/portrait-mode-now-available-on-iphone-7-plus-with-ios-101/</p>
[21d.] to apply a different type or	The '132 Patent Accused Products include a correction unit to apply a different type or amount of correction to a second portion of the

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
<p>amount of correction to a second portion of the set of digital image data which does not represent a physical object of the predetermined type.</p>	<p>set of digital image data which does not represent a physical object of the predetermined type.</p> <p>See [21c].</p> <p>See, e.g.,:</p> <p>Core Image is an image processing and analysis technology that provides high-performance processing for still and video images.</p> <p>https://developer.apple.com/documentation/coreimage.</p> <p>Core Image can analyze and find human faces in an image. It performs face detection, not recognition. Face detection is the identification of rectangles that contain human face features, whereas face recognition is the identification of specific human faces (John, Mary, and so on). After Core Image detects a face, it can provide information about face features, such as eye and mouth positions. It can also track the position an identified face in a video.</p> <p>Figure 2-1 Core Image identifies face bounds in an image</p>  <p>Knowing where the faces are in an image lets you perform other operations, such as cropping or adjusting the image quality of the face (tone balance, red-eye correction and so on). You can also perform other interesting operations on the faces; for example:</p> <ul style="list-style-type: none"> • <i>Anonymous Faces Filter Recipe</i> shows how to apply a pixelate filter only to the faces in an image. • <i>White Vignette for Faces Filter Recipe</i> shows how to place a vignette around a face. <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>White Vignette for Faces Filter Recipe</p> <p>Increases the brightness of an image at the periphery of a face detected in an image.</p> <p>Figure 5-2 The White Vignette filter processing chain</p>  <p>To create a white vignette filter:</p> <ul style="list-style-type: none">• Find the human face in an image.• Create a base shade map using CIRadialGradient centered on the face.• Blend the base shade map with the original image. <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filter_recipes/ci_filter_recipes.html#//apple_ref/doc/uid/TP30001185-CH4-SW22</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p data-bbox="513 737 803 762">Anonymous Faces Filter Recipe</p> <p data-bbox="513 783 878 798">Finds faces in an image and pixellates them so they can't be recognized.</p> <p data-bbox="545 821 836 833">Figure 5-4 The Anonymous Faces filter processing chain</p>  <p data-bbox="513 1150 698 1165">To create an anonymous faces filter:</p> <ul data-bbox="513 1173 862 1234" style="list-style-type: none">• Create a pixelated version of the image.• Build a mask using the faces detected in the image.• Blend the pixelated image with the original image using the mask. <p data-bbox="448 1264 1515 1308">https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filter_recipes/ci_filter_recipes.html#//apple_ref/doc/uid/TP30001185-CH4-SW22</p> <p data-bbox="506 1331 1070 1417">Build a Mask From the Faces Detected in the Image Use the CIDetector class for find the faces in the image. For each face: Use the CIRadialGradient filter to create a circle that surrounds the face. Use the CISourceOverCompositing filter to add the gradient to the mask.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filter_recipes/ci_filter_recipes.html#apple_ref/doc/uid/TP30001185-CH4-SW22</p> <p>Blend the Pixellated Image, the Mask, and the Original Image Set the input parameters of the CIBlendWithMask filter to the following: Set inputImage to the pixellated version of the image. Set inputBackgroundImage to the original image. Set inputMaskImage to the composited green circles.</p> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filter_recipes/ci_filter_recipes.html#apple_ref/doc/uid/TP30001185-CH4-SW22</p> <p>Getting Face and Face Feature Bounds</p> <p>Face features include:</p> <ul style="list-style-type: none"> • left and right eye positions • mouth position • tracking ID and tracking frame count which Core Image uses to follow a face in a video segment (available in iOS v6.0 and later and in OS X v10.8 and later) <p>After you get an array of face features from a CIDetector object, you can loop through the array to examine the bounds of each face and each feature in the faces, as shown in Listing 2-2.</p>


Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Listing 2-2 Examining face feature bounds</p> <pre> for (CIFaceFeature *f in features) { NSLog(@"%@", NSStringFromRect(f.bounds)); if (f.hasLeftEyePosition) { NSLog(@"Left eye %g %g", f.leftEyePosition.x, f.leftEyePosition.y); } if (f.hasRightEyePosition) { NSLog(@"Right eye %g %g", f.rightEyePosition.x, f.rightEyePosition.y); } if (f.hasMouthPosition) { NSLog(@"Mouth %g %g", f.mouthPosition.x, f.mouthPosition.y); } } </pre> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html</p> <p>Create a Shade Map</p> <p>Use the CIRadialGradient filter to create a shade map centered on the face. The center of the shade map should be transparent so that the face in the image remains untouched. The edges of the map should be opaque white. Areas in between should have varying degrees of transparency.</p> <p>To achieve this effect, set the input parameters to CIRadialGradient as follows:</p> <ul style="list-style-type: none"> • Set inputRadius0 to a value larger than the longest dimension of the the image. • Set inputRadius1 to a value larger than the face, such as face.bounds.size.height + 50. • Set inputColor0 to opaque white. • Set inputColor1 to transparent white. • Set the inputCenter to the center of the face bounds that you computed with Listing 5-4. <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_filter_recipes/ci_filter_recipes.html#apple_ref/doc/uid/TP30001185-CH4-SW22</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))												
	<p>Table 3-1 Filters that Core Image uses to enhance an image</p> <table border="1"> <thead> <tr> <th>Filter</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>CIRedEyeCorrection</td><td>Repairs red/amber/white eye due to camera flash</td></tr> <tr> <td>CIFaceBalance</td><td>Adjusts the color of a face to give pleasing skin tones</td></tr> <tr> <td>CIVibrance</td><td>Increases the saturation of an image without distorting the skin tones</td></tr> <tr> <td>CIToneCurve</td><td>Adjusts image contrast</td></tr> <tr> <td>CIHighlightShadowAdjust</td><td>Adjusts shadow details</td></tr> </tbody> </table> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html#//apple_ref/doc/uid/TP30001185-CH11-SW1.</p>	Filter	Purpose	CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash	CIFaceBalance	Adjusts the color of a face to give pleasing skin tones	CIVibrance	Increases the saturation of an image without distorting the skin tones	CIToneCurve	Adjusts image contrast	CIHighlightShadowAdjust	Adjusts shadow details
Filter	Purpose												
CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash												
CIFaceBalance	Adjusts the color of a face to give pleasing skin tones												
CIVibrance	Increases the saturation of an image without distorting the skin tones												
CIToneCurve	Adjusts image contrast												
CIHighlightShadowAdjust	Adjusts shadow details												

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))				
	<div><div>CIVibrance</div><div>Adjusts the saturation of an image while keeping pleasing skin tones.</div><div>Localized Display Name Vibrance</div><div>Parameters<table><tr><td>inputImage</td><td>A CIImage object whose display name is Image.</td></tr><tr><td>inputAmount</td><td>An NSNumber object whose attribute type is CIColorAdjustment and whose display name is Amount.</td></tr></table></div><div>Member Of CIColorAdjustment, CIColorAdjustmentNonSquarePixels, CIColorAdjustmentInterlaced, CIColorAdjustmentStillImage, CIColorAdjustmentVideo, CIColorAdjustmentColorAdjustment</div><div>Sample Output Figure 20 The result of using the CIVibrance filter </div><div>Availability Available in OS X v10.7 and later and in iOS 5.0 and later.</div></div> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Reference/CoreImageFilterReference/index.html#//apple_ref/doc/filter/ci/CIVibrance.</p>	inputImage	A CIImage object whose display name is Image.	inputAmount	An NSNumber object whose attribute type is CIColorAdjustment and whose display name is Amount.
inputImage	A CIImage object whose display name is Image.				
inputAmount	An NSNumber object whose attribute type is CIColorAdjustment and whose display name is Amount.				

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>As another example, the iPhone 7Plus, iPhone 8Plus, and iPhone X each can operate in Portrait mode. When using Portrait mode, the iPhone 7Plus, iPhone 8Plus, and iPhone X can apply a correction to the second portion of the set of digital image data, <i>i.e.</i> portions of the image other than the faces of the subjects.</p> <p>Choose a photo mode. Camera has several photo modes so you can shoot standard and square-format photos, and panoramas. To choose a mode, swipe the screen left or right, or tap Photo, Square, or Pano.</p> <p>On iPhone X, iPhone 8 Plus, and iPhone 7 Plus, you can also choose Portrait mode, which uses a depth effect to focus on your subject and blur the background.</p> <p>iPhone User Guide For iOS 11.2 software at Chapter "Camera," available at https://help.apple.com/iphone/11/.</p>

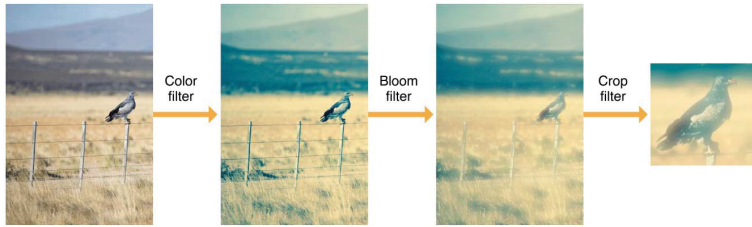
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div data-bbox="867 737 1097 1199" data-label="Image"> </div> <p data-bbox="448 1220 1377 1245">[Portrait mode] lets you compose a photo that keeps your subject sharp while creating a beautifully blurred background.</p> <p data-bbox="506 1264 839 1287">https://support.apple.com/en-us/HT208118</p> <p data-bbox="448 1308 1468 1354">See also iPhone User Guide For iOS 11.2 software at Chapter “Camera,” available at https://help.apple.com/iphone/11/ (discussing different lighting effects that can be applied to the subject of a picture shot in Portrait mode).</p> <p data-bbox="506 1375 583 1398">Overview</p>

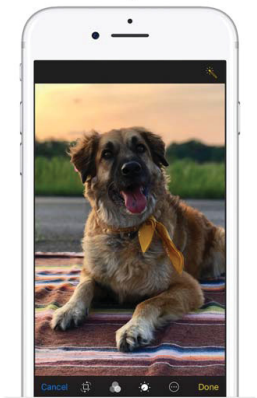
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>The CIFilterGenerator class provides methods for creating a CIFilter object by chaining together existing CIFilter objects to create complex effects. (A filter chain refers to the CIFilter objects that are connected in the CIFilterGenerator object.) The complex effect can be encapsulated as a CIFilterGenerator object and saved as a file so that it can be used again. The filter generator file contains an archived instance of all the CIFilter objects that are chained together.</p> <p>Any filter generator files that you copy to /Library/Graphics/Image Units/ are loaded when any of the loading methods provided by the CIPugin class are invoked. A CIFilterGenerator object is registered by its filename or, if present, by a class attribute that you supply in its description.</p> <p>You can create a CIFilterGenerator object programmatically, using the methods provided by the CIFilterGenerator class, or by using the editor view provided by Core Image.</p> <p>Core Image Documentation, CIFilterGenerator, https://developer.apple.com/documentation/coreimage/cifiltergenerator.</p> <p>sourceKey The key that specifies the source object. For example, if the source is the output image of a filter, pass the outputImage key. Pass nil if the source object is used directly.</p> <p>Core Image Documentation, connect(:withKey:to:withKey:), https://developer.apple.com/documentation/coreimage/cifiltergenerator/1438159-connect.</p> <p>Core Image filters process and produce Core Image images. A CIImage instance is an immutable object representing an image. These objects don't directly represent image bitmap data—instead, a CIImage object is a “recipe” for producing an image. One recipe might call for loading an image from a file; another might represent output from a filter, or from a chain of filters. Core Image performs these recipes only when you request that an image be rendered for display or output.</p> <p>Core Image Programming Guide, Processing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_tasks/ci_tasks.html.</p> <p>Chaining Filters for Complex Effects Every Core Image filter produces an output CIImage object, so you can use this object as input to another filter. For example, the</p>

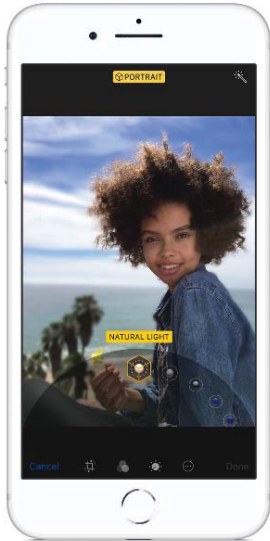
Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>sequence of filters illustrated in Figure 1-1 applies a color effect to an image, then adds a glow effect, and finally crops a section out of the result.</p> <p>Figure 1-1 Construct a filter chain by connecting filter inputs and outputs</p>  <p>Core Image Programming Guide, Processing Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_tasks/ci_tasks.html.</p> <p>Additionally, correction may be determined and applied by extensions. <i>See, e.g.,</i>:</p> <p>Third-party apps and extensions expand your editing options in Photos for iOS and macOS. You can apply edits from multiple apps and extensions to one photo, or use any combination of apps and extensions plus the editing tools built into Photos.</p> <p>https://support.apple.com/en-us/HT205245</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Enhance instantly</p>  <p>Auto-enhance will instantly improve your picture's exposure, contrast, saturation, and other qualities. It's the fastest way to make your photo even more amazing.</p> <p>https://support.apple.com/en-us/ht205659</p> <p>Additionally, the iPhone 7 Plus, iPhone 8 Plus, and iPhone X support Portrait mode for detecting faces and applying corrections:</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div></div> <div><h3>Add Portrait Lighting</h3><p>On iPhone 8 Plus and iPhone X, you can use the Portrait Lighting feature to apply four impressive studio-quality lighting effects to your Portrait mode images. Choose from Studio Light to brighten facial features, Contour Light for more dramatic directional lighting, Stage Light to isolate your subject in the spotlight, or Stage Mono for stage light in a classic black and white.</p><ol style="list-style-type: none">1. Select the photo that you want to change.2. Tap Edit.3. Swipe to choose your Lighting effect.4. Tap Done.<p>Portrait Lighting is currently in beta.</p></div> <p>https://support.apple.com/en-us/HT208118; see also https://www.apple.com/newsroom/2016/10/portrait-mode-now-available-on-iphone-7-plus-with-ios-101/</p>
Claim 22	

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))												
[22a.] An image processing unit as recited in claim 21,	The '132 Patent Accused Products are the image processing unit as recited in claim 21. <i>See</i> [21a]-[21d].												
[22b.] wherein the correction is a dynamic range correction.	<p>The '132 Patent Accused Products perform dynamic range correction.</p> <p><i>See</i> [21c]-[21d].</p> <p><i>See, e.g.,</i>:</p> <p>Core Image is an image processing and analysis technology that provides high-performance processing for still and video images.</p> <p>https://developer.apple.com/documentation/coreimage.</p> <p>Table 3-1 Filters that Core Image uses to enhance an image</p> <table border="1" data-bbox="514 1045 1289 1316"> <thead> <tr> <th>Filter</th><th>Purpose</th></tr> </thead> <tbody> <tr> <td>CIRedEyeCorrection</td><td>Repairs red/amber/white eye due to camera flash</td></tr> <tr> <td>CIFaceBalance</td><td>Adjusts the color of a face to give pleasing skin tones</td></tr> <tr> <td>CIVibrance</td><td>Increases the saturation of an image without distorting the skin tones</td></tr> <tr> <td>CIToneCurve</td><td>Adjusts image contrast</td></tr> <tr> <td>CIHighlightShadowAdjust</td><td>Adjusts shadow details</td></tr> </tbody> </table> <p>https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_autoadjustment/ci_autoadjustmentSAVE.html#//apple_ref/doc/uid/TP30001185-CH11-SW1</p>	Filter	Purpose	CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash	CIFaceBalance	Adjusts the color of a face to give pleasing skin tones	CIVibrance	Increases the saturation of an image without distorting the skin tones	CIToneCurve	Adjusts image contrast	CIHighlightShadowAdjust	Adjusts shadow details
Filter	Purpose												
CIRedEyeCorrection	Repairs red/amber/white eye due to camera flash												
CIFaceBalance	Adjusts the color of a face to give pleasing skin tones												
CIVibrance	Increases the saturation of an image without distorting the skin tones												
CIToneCurve	Adjusts image contrast												
CIHighlightShadowAdjust	Adjusts shadow details												


Qualcomm Inc.’s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>Choose a photo mode. Camera has several photo modes so you can shoot standard and square-format photos, and panoramas. To choose a mode, swipe the screen left or right, or tap Photo, Square, or Pano.</p> <p>On iPhone X, iPhone 8 Plus, and iPhone 7 Plus, you can also choose Portrait mode, which uses a depth effect to focus on your subject and blur the background.</p> <p>iPhone User Guide For iOS 11.2 software at Chapter “Camera,” available at https://help.apple.com/iphone/11/.</p> <div data-bbox="867 905 1097 1367" data-label="Image"> </div> <p>[Portrait mode] lets you compose a photo that keeps your subject sharp while creating a beautifully blurred background.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	https://support.apple.com/en-us/HT208118 <i>See also</i> iPhone User Guide For iOS 11.2 software at Chapter "Camera," available at https://help.apple.com/iphone/11/ (discussing different lighting effects that can be applied to the subject of a picture shot in Portrait mode). Each of these corrections is a dynamic range correction.
Claim 23	
[23a.] An image processing unit as recited in claim 22,	The '132 Patent Accused Products are the image processing unit as recited in claim 22. <i>See</i> [22a]-[22b].
[23b.] wherein the object detection unit is configured to detect faces.	The '132 Patent Accused Products include an object detection unit that is configured to detect faces. <i>See</i> [21b].


Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<div data-bbox="508 737 846 779">  </div> <p data-bbox="508 800 812 1163"> Both models: Optical image stabilization Six-element lens Quad-LED True Tone flash Panorama (up to 63MP) Sapphire crystal lens cover Backside illumination sensor Hybrid IR filter Autofocus with Focus Pixels Tap to focus with Focus Pixels Live Photos with stabilization Wide color capture for photos and Live Photos Improved local tone mapping Body and face detection </p> <p data-bbox="448 1171 1193 1194"> See https://www.apple.com/iphone-7/specs/ (identifying “body and face detection” as a feature). </p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>iPhone 8 ▼</p> <p>Both models:</p> <p>Optical image stabilization</p> <p>Six-element lens</p> <p>Quad-LED True Tone flash with Slow Sync</p> <p>Panorama (up to 63MP)</p> <p>Sapphire crystal lens cover</p> <p>Backside illumination sensor</p> <p>Hybrid IR filter</p> <p>Autofocus with Focus Pixels</p> <p>Tap to focus with Focus Pixels</p> <p>Live Photos with stabilization</p> <p>Wide color capture for photos and Live Photos</p> <p>Improved local tone mapping</p> <p>Body and face detection</p> <p>https://www.apple.com/iphone-8/specs/ (identifying “body and face detection” as a feature of the iPhone 8 and the iPhone 8 Plus);</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>iPhone X <small>aperture: f/1.8 aperture</small> </p> <hr/> <p>Telephoto: f/2.4 aperture</p> <p>Optical zoom; digital zoom up to 10x</p> <p>Portrait mode</p> <p>Portrait Lighting (beta)</p> <p>Dual optical image stabilization</p> <p>Six-element lens</p> <p>Quad-LED True Tone flash with Slow Sync</p> <p>Panorama (up to 63MP)</p> <p>Sapphire crystal lens cover</p> <p>Backside illumination sensor</p> <p>Hybrid IR filter</p> <p>Autofocus with Focus Pixels</p> <p>Tap to focus with Focus Pixels</p> <p>Live Photos with stabilization</p> <p>Wide color capture for photos and Live Photos</p> <p>Improved local tone mapping</p> <p>Body and face detection</p> <p>https://www.apple.com/iphone-x/specs/ (identifying “body and face detection” as a feature of the iPhone X).</p> <p>var landmarks: VNFaceLandmarks2D?</p> <p>The facial features found in the detected face.</p> <p>class VNFaceLandmarks2D</p> <p>A collection of all facial features detected in a face observation, with 2D geometry information for each.</p> <p>class VNFaceLandmarkRegion2D</p> <p>2D geometry information for a specific facial feature.</p> <p>class VNFaceLandmarks</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>The abstract superclass for containers of face landmark information. class VNFaCeLandmarkRegion The abstract superclass for information about a specific face landmark.</p> <p>Vision Framework for iOS 11.0+, VNFaCeObservation, https://developer.apple.com/documentation/vision/vnfaceobservation.</p> <p>Core Image can detect human face features in still images and track them over time in video images. Knowing where faces are can help you determine where to place a vignette or apply other special filters.</p> <p>Core Image Programming Guide, About Core Image, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_intro/ci_intro.html#//apple_ref/doc/uid/TP30001185.</p> <p>Core Image can analyze and find human faces in an image. It performs face detection, not recognition. Face detection is the identification of rectangles that contain human face features, whereas face recognition is the identification of specific human faces (John, Mary, and so on). After Core Image detects a face, it can provide information about face features, such as eye and mouth positions. It can also track the position an identified face in a video.</p> <p>Knowing where the faces are in an image lets you perform other operations, such as cropping or adjusting the image quality of the face (tone balance, red-eye correction and so on). You can also perform other interesting operations on the faces; for example:</p> <p>Anonymous Faces Filter Recipe shows how to apply a pixellate filter only to the faces in an image.</p> <p>White Vignette for Faces Filter Recipe shows how to place a vignette around a face.</p> <p>Core Image Programming Guide, Detecting Faces in Images, https://developer.apple.com/library/content/documentation/GraphicsImaging/Conceptual/CoreImaging/ci_detect_faces/ci_detect_faces.html#//apple_ref/doc/uid/TP30001185-CH8-SW1.</p> <p>Model barcode data to create barcode images with Core Image filters or to interpret barcodes detected by Core Image, Vision, or</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>AVFoundation features.</p> <p>class CIBarcodeDescriptor</p> <p>An abstract base class that represents a machine readable code's attributes.</p> <p>class CIQRCodeDescriptor</p> <p>A concrete subclass of CIBarcodeDescriptor that represents a square QR code symbol.</p> <p>class CIAztecCodeDescriptor</p> <p>A concrete subclass of CIBarcodeDescriptor that represents an Aztec code symbol.</p> <p>class CIPDF417CodeDescriptor</p> <p>A concrete subclass of CIBarcodeDescriptor that represents a PDF 417 symbol.</p> <p>class CIDataMatrixCodeDescriptor</p> <p>A concrete subclass of CIBarcodeDescriptor that represents a Data Matrix code symbol.</p> <p>Core Image Framework Documentation, https://developer.apple.com/documentation/coreimage.</p> <p>class CIDetector</p> <p>An image processor that identifies notable features (such as faces and barcodes) in a still image or video.</p> <p>class CIFeature</p> <p>The abstract superclass for objects representing notable features detected in an image.</p> <p>class CIFaceFeature</p> <p>Information about a face detected in a still or video image.</p> <p>class CIRectangleFeature</p> <p>Information about a rectangular region detected in a still or video image.</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim ² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found ³ (P.R. 3-1(b)-(c))
	<p>class CITextFeature Information about a region likely to contain text detected in a still or video image. class CIQRCodeFeature Information about a Quick Response code (a kind of 2D barcode) detected in a still or video image.</p> <p>Core Image Framework Documentation, https://developer.apple.com/documentation/coreimage.</p> <p>Overview</p> <p>A CIDetector object uses image processing to search for and identify notable features (faces, rectangles, and barcodes) in a still image or video. Detected features are represented by CIFeature objects that provide more information about each feature.</p> <p>Core Image Framework Documentation, https://developer.apple.com/documentation/coreimage/cidetector.</p>
Claim 33	
[33a.] An apparatus comprising:	The '132 Patent Accused Products are an apparatus. <i>See</i> [33b]-[33e].
[33b.] means for making a determination that a first group of adjacent pixels in a digital image represents a physical object of a predetermined type;	<p>The '132 Patent Accused Products include means for making a determination that a first group of adjacent pixels in a digital image represents a physical object of a predetermined type.</p> <p><i>See</i> [21b].</p>
[33c.] means for determining a type or amount of dynamic range correction to apply to at least one pixel in the first group of adjacent pixels based on the determination that the first group of adjacent pixels represents a physical object of the predetermined type;	<p>The '132 Patent Accused Products includemeans for determining a type or amount of dynamic range correction to apply to at least one pixel in the first group of adjacent pixels based on the determination that the first group of adjacent pixels represents a physical object of the predetermined type.</p> <p><i>See</i> [21c] and [22b].</p>

Qualcomm Inc.'s Patent Initial Infringement Contentions
Exhibit 132

'132 Patent Claim² (P.R. 3-1(a))	The '132 Patent Accused Products And Where Each Claim Element Is Found³ (P.R. 3-1(b)-(c))
[33d.] means for applying the determined type or amount of dynamic range correction to the at least one pixel in the first group of adjacent pixels, by applying a first amount of the correction to the first portion of the digital image data; and	The '132 Patent Accused Products include means for applying the determined type or amount of dynamic range correction to the at least one pixel in the first group of adjacent pixels, by applying a first amount of the correction to the first portion of the digital image data. See [21c] and [22b].
[33e.] means for applying a second amount of the correction to a second portion of the digital image data, wherein the first amount differs from the second amount, and wherein the first amount corresponds to a physical object of the predetermined type.	The '132 Patent Accused Products include means for applying a second amount of the correction to a second portion of the digital image data, wherein the first amount differs from the second amount, and wherein the first amount corresponds to a physical object of the predetermined type. See [21d].