

DESCRIPTION

Title of Invention: IMAGE PROCESSING METHOD, IMAGE PROCESSING DEVICE

Technical Field

[0001]

The present disclosure relates to a technique for processing images using neural network.

Background Art

[0002]

Object detection plays an important role in many image processing tasks. The motivation behind object detection is to find a certain object or a part of it in an image. After the object is localized, other tasks such as for example object identification can be performed.

[0003]

One of embodiments of object detection is face recognition. Face recognition generally requires faces to be aligned in a certain orientation to achieve high recognition performance. For the alignment of this face region, the face's features such as eyes, nose, mouth, etc. can be used. With object detection these facial features can be identified. It is possible to align the orientation of the face using the identified facial features.

[0004]

In object detection, so-called sliding windows are used. This approach extracts an arbitrary region from the image, and calculates the probability by which the extracted region includes a certain object. If the probability is beyond a given threshold, the object is detected in the extracted region. With this sliding window approach high accuracy can be achieved. However, this approach has a drawback of high computational costs, if a-priori information about possible object locations is not available. This is because the window must be placed at many different locations and scaled until reaching the pre-defined probability threshold.

[0005]

Furthermore, this sliding window approach often uses template matching to calculate the probability of a certain object being in a region. These templates are often pre-defined and do not adapt well, if the object appearance changes over time. Therefore, it is necessary to update the template to achieve high detection accuracy.

[0006]

Machine learning is a method for adapting changes of the object. Machine learning approach abstracts external features of the object to identify such features by learning external characteristic elements identifying the object. In general, machine learning previously learns known pairs of image data and known detected result acquired from the image data. The learner performs object detection to new image data using the learned result. It is also possible to adjust learning parameters depending on accuracy of detected results.

[0007]

Neural network is commonly known as a learner implementing machine learning. Neural networks include artificial neurons inspired by the neurons in a human brain. A given weighting factor is multiplied with inputs to each neuron, and then a bias is added, thereby acquiring outputs of the neuron.

[0008]

Patent Literature 1 listed below describes a method for identifying humans in a low resolution image using a convolution neural network. In Patent Literature 1, the learning process is repeated using images of the human to be identified.

Citation List

Patent Literature

[0009]

Patent Literature 1: US Patent Publication 2011/0222724

Summary of Invention

Technical Problem

[0010]

When performing object detection using neural network, weighting factors and biases are adjusted in the learning process to acquire desired detection results. One of technical problems when performing object detection using neural network is that long learning time is necessary until acquiring desired results.

[0011]

The present disclosure has been provided in view of the above problems. It is an objective of the present disclosure to improve efficiency of the process for detecting objects using neural network.

Solution to Problem

[0012]

The present disclosure : converts tasks required for object detection into tasks required for solving classification problem ; the object detection is handled as a classification problem using neural network that classifies inputted data.

Advantageous Effects of Invention

[0013]

With the present disclosure, it is possible to efficiently handle tasks required in object detection as classification problem by using neural network that classifies inputted data. Technical problems, configurations, and effects other than above will be apparent from descriptions of embodiments below.

Brief Description of Drawings

[0014]

Figure 1 is a functional block diagram of an image processing device 100 according to an embodiment 1.

Figure 2 is a diagram showing a process of a region definer 110, a process of a coordinate separator 120, and a process of a quantizer 130.

Figure 3 is a diagram showing an example of layer configuration of a learner 150.

Figure 4 is a diagram explaining an output vector outputted from a classifier 160.

Figure 5 is a diagram showing a process of a reproducer 170.

Figure 6 is a functional block diagram of the image processing device 100 according to an embodiment 2.

Figure 7 is a diagram showing a process of a shifter 180.

Figure 8 is a diagram showing a process of the region definer 110 and a process of the quantizer 130 when image data 200 is an image of a running car captured from a side of the car.

Figure 9 is a diagram showing a process of the region definer 110 when image data 200 is an image of a car captured from front of the car.

Figure 10 is a diagram showing hardware configuration of the image processing device 100 according to an embodiment 4.

Description of Embodiments

[0015]

<Basic concept of the disclosure>

Hereinafter, for the sake of better understanding of the present disclosure, a basic concept of the present disclosure will be described firstly. Then embodiments of the present disclosure will be described.

[0016]

Convolution neural network described in Patent Literature 1 is one of methods for detecting objects using neural network. In the method of Patent Literature 1, for example : a shape detection filter is applied (convolution) to image data, thereby detecting the shape of the object ; some method such as Max Pooling is applied to the output from the filter, thereby diluting the positional error of the object in the filter. The neural network is formed by connecting a plurality of layers implementing the processing above.

[0017]

Other application of neural network is to solve classification problem. In this case, the neural network previously learns the relationship between the input data and the

classification to which the input data belongs. When entering new input data into the neural network, the neural network outputs a value showing a classification to which the new input data belongs, according to the previously learned result.

[0018]

In general, neural network can solve classification problems more efficiently than performing object detections. This is because : object detection is a task to identify a position where the object exists ; whereas classification problem is a task to answer the classification to which the classified target belongs, in which the outputted result is more simplified than that of object detection. The present disclosure focuses on this nature. The present disclosure intends to improve the processing efficiency for detecting objects using neural network by providing measures for handling object detection problem as classification problem.

[0019]

<Embodiment 1>

Figure 1 is a functional block diagram of an image processing device 100 according to an embodiment 1 of the present disclosure. The image processing device 100 is a device that detects objects in image data 200 using neural network. The image processing device 100 includes a region definer 110, a coordinate separator 120, a quantizer 130, a mapper 140, a learner 150, a classifier 160, and a reproducer 170. Hereinafter, facial images of human will be handled as an example of the image data 200. Behaviors of each functional unit will be described later.

[0020]

Figure 2 is a diagram showing a process of the region definer 110, a process of the coordinate separator 120, and a process of the quantizer 130. Hereinafter, behaviors of these three functional units will be described according to Figure 2.

[0021]

The region definer 110 defines one or more sub regions on the image data 200 depending on numbers and positions of characteristic elements that are assumed to be included in the image data 200. When detecting a face of human, for example : the left top region (when seeing in the direction toward the figure) of the image data 200 is defined as a right eye

region ; the right top region is defined as a left eye region ; and the lower half is defined as a mouth region. The number and the position of the sub regions defined by the region definer 110 may be defined in advance depending on the characteristics of the detected object. Alternatively, the number and the position of the sub regions may be designated every time via some appropriate interface included in the image processing device 100. It is not always necessary to define the sub regions. For example, whole of the image data 200 may be defined as one region to perform object detection. Hereinafter, it is assumed that the sub regions are defined as shown in Figure 2.

[0022]

The coordinate separator 120 separates pixel coordinates of each of the sub regions into values of each coordinate axes. Each of coordinates after separated is handled as individual one dimensional vector. Since the embodiment 1 handles two dimensional images of human face, each pixel coordinates will be separated into X coordinate values and Y coordinate values.

[0023]

The quantizer 130 quantizes pixel coordinate values of each coordinate axis separated by the coordinate separator 120 using a predetermined interval. Then pixels in each sub region are converted from continuous pixels at each of one pixel into discrete pixels at each of the interval. The quantization interval for each sub region is not necessarily identical to each other. Further, the quantization interval for each coordinate axis is not necessarily identical to each other. As shown in Figure 2, for example, the quantization interval in a small region may be set to be small, and the quantization interval in a horizontally long region may be set so that the quantization interval of X coordinate is larger than that of Y coordinate. The quantization interval defines the error width when the reproducer 170 mentioned below reproduces each of characteristic elements on the coordinate system. Details will be described later. The quantizer 130 is for reducing computational loads, and thus the quantizer 130 is not necessary when sufficient computational resource is provided.

[0024]

The mapper 140 performs processes required for handling the object detection

problem as a classification problem. When handling the object detection problem as a classification problem, it can be assumed that : each of layers formed in the convolution neural network classifies whether each of pixels in the image data 200 corresponds to a coordinate value of the target object ; and finally the coordinate value of the target object is acquired by aggregating outputs from each layer. Then the present disclosure assumes that each of pixel coordinate values in the image data 200 (if quantization is performed, the coordinate values after quantized) is a classified target in a classification problem. The present disclosure then classifies each of the coordinate values into ones belonging to the target object and ones not belonging to the target object (i.e. whether each of the coordinate values is included in a part of the target object).

[0025]

The mapper 140 assigns a classification in the learner 150 to the coordinate values (if quantization is performed, the coordinate values after quantized) of each pixel in the image data 200. The relationship between the pixel coordinate values and the internal classifications may be defined in advance as a conversion table, for example. The conversion from the pixel coordinate values into the internal classifications of the learner 150 is performed for each of the sub regions and for each of the coordinate axes. In the embodiment 1, there are three sub regions and XY axes. Thus the mapper 140 converts XY pixel coordinates of the right eye region / the left eye region / the mouth region into the internal classifications of the learner 150.

[0026]

The learner 150 is a machine learner that is formed using such as convolution neural network. The learner 150 previously learns whether each of pixel coordinates classified by the mapper 140 is a coordinate value of the detection target object. In the embodiment 1, the detection target objects are the right eye / the left eye / the mouth in the image data 200. The learner 150 identifies the coordinates of the right eye / the left eye / the mouth of new one of the image data 200 using the previously learned result.

[0027]

Figure 3 is a diagram showing an example of layer configuration of the learner 150.

It is assumed in the figure that the image data 200 is a two dimensional image of $64 \text{ pixel} \times 64$ pixels. Hereinafter, a configuration example of the learner 150 will be described using Figure 3.

[0028]

The first layer applies, to the image data 200, a shape detection filter of $9 \text{ pixels} \times 9$ pixels. If this filter is a filter that detects vertical lines, for example, this filter is a 9×9 array with the center column having large values and with other columns having small values. The learner 150 aligns the left-top pixel of the image data 200 with the left-top edge of the filter. Then the learner 150 multiplies each of the pixel values in the filter window with the filter value at the same position as the pixel. It is further possible to add bias values to the multiplied result depending on design conditions. The bias value may be same for all elements in the filter. Alternatively, the bias value may be different for each of elements. The learner 150 sums up the results acquired by multiplying each of the pixel values with the filter value (or the result acquired by further adding the bias values) for each of 9×9 elements in the filter. The learner 150 inputs the summed result to the neuron corresponding to the pixel position.

[0029]

The neuron receiving the input value inputs the input value to a predetermined activation function to calculate an output value. Examples of the activation function may be nonlinear filtering functions such as Rectified Linear Unit (ReLU). ReLU is defined as $f(x) = \max(0, x)$.

[0030]

The learner 150 moves the filter window from the left-top pixel of the image data 200 sequentially, thereby scanning all pixels of the image data 200 by the filter. The learner 150 performs same calculations as above at each position of the filter window, and inputs the calculation result to the neuron corresponding to the position. The neuron calculates output values. In the example shown in Figure 3, the filter of 9×9 pixels is applied to the image data 200 of 64×64 pixels. Thus the start positions of the filter are from the first pixel to the 56th pixel. Therefore, processing results are acquired corresponding to 56×56 pixels after

the first layer.

[0031]

If only one type of the shape detection filter is used, then only one type of shape is detectable. If a plurality of shapes is to be detected, a plurality of types of shape detection filters may be applied at each layer. In Figure 3, 20 types of shape detection filters are applied at the first layer, and neurons are provided for each filter type. This type of method where a plurality of shape detection filters is provided for each layer is referred to as Feature Map. Consequently, the first layer outputs processing results corresponding to $56 \times 56 \times 20$ pixels.

[0032]

When applying shape detection filters in convolution neural network, the position of shape of detection target may be shifted from a position that is assumed to exist in the filter window. Some method such as Max Pooling may be used to dilute such positional errors. Max Pooling is a method where only the maximum value in the window is kept. The shape in the window is abstracted by the maximum value. This corresponds to reducing the resolution of the image. In the example shown in Figure 3, Max Pooling is performed using a 2×2 window to the output from the first layer, thereby acquiring a processing result of $28 \times 28 \times 20$.

[0033]

The second layer applies a 7×7 filter window to the processing result of $28 \times 28 \times 20$ acquired by Max Pooling. Since higher layers detect more complicated shapes than lower layers, the number of shape detection filters at the second layer is 40. Subsequent layers perform same processing as previous layers. In the configuration example shown in Figure 3, a processing result of $3 \times 3 \times 60$ is finally acquired.

[0034]

The fully connected layer at the final stage has 120 neurons. All neurons in previous layers are connected to some neurons in the fully connected layer. Therefore, the fully connected layer has information of all neurons in previous layers.

[0035]

Figure 4 is a diagram explaining an output vector outputted from the classifier 160. The classifier 160 performs logistic regression analysis to each output from 120 neurons included in the fully connected layer. The classifier 160 then outputs analysis results regarding right eye X coordinate / right eye Y coordinate / left eye X coordinate / left eye Y coordinate / mouth X coordinate / mouth Y coordinate. This analysis result may be handled as a vector having six elements.

[0036]

The learner 150 has internal classifications corresponding to XY coordinates of right eye region / left eye region / mouth region, respectively. Thus it can be assumed that the classifications included in the learner 150 are aggregated into six classifications of right eye X coordinate / right eye Y coordinate / left eye X coordinate / left eye Y coordinate / mouth X coordinate / mouth Y coordinate, as the information is aggregated from lower layers to higher layers. Then in the embodiment 1, the classifier 160 acquires output vectors having these six element values by performing logistic regression analysis to output values from each neuron in the fully connected layer at the last stage of the learner 150.

[0037]

Figure 5 is a diagram showing a process of the reproducer 170. The reproducer 170 reproduces six coordinates of right eye X coordinate / right eye Y coordinate / left eye X coordinate / left eye Y coordinate / mouth X coordinate / mouth Y coordinate on the coordinate system in the image data 200, using output vectors from the classifier 160. Specifically, (a) each element value of the output vector is projected onto the coordinate system in the image data 200 using the quantizer 130, (b) each of the inverse-quantized element values is formatted into XY coordinates using the coordinate separator 120, (c) each of the XY coordinates is assigned to sub regions in the image data 200 using the region definer 110.

[0038]

Each element value of the output vector corresponds to quantized XY coordinates of right eye / left eye / mouth. The quantizer 130 inverse-quantizes the quantized XY coordinates into continuous coordinate values. For example, it is assumed that the quantizer 130 is configured to quantize coordinate values from at or above 0.3 to below 0.4 in the image

data 200 into 0.3. If an element value of output vector is 0.3, the quantizer 130 inverse-quantizes the element value into a coordinate value from at or above 0.3 to below 0.4. Any coordinate value from at or above 0.3 to below 0.4 is acceptable. In order to minimize the difference between the inverse-quantized coordinate value and the true XY coordinates of right eye / left eye / mouth, it is desirable to inverse-quantize the coordinate value into 0.35, a center value. Then the expected difference will be minimized between the inverse-quantized coordinate value and any one of true coordinate value from at or above 0.3 to below 0.4.

[0039]

The coordinate separator 120 generates (X, Y) coordinate by pairing the inverse-quantized XY coordinates. For example, by pairing an inverse-quantized logistic regression analysis result of right eye X coordinate with an inverse-quantized logistic regression analysis result of right eye Y coordinate, a coordinate vector of right eye (X_{RE} , Y_{RE}) will be generated. It also applies to left eye and mouth.

[0040]

The region definer 110 assigns the XY coordinates of right eye / left eye / mouth acquired by the coordinate separator 120 to corresponding each sub region.

[0041]

As a result of above, XY coordinates of right eye / left eye / mouth will be reproduced on the coordinate system of the image data 200. The lower diagram in Figure 5 shows an example of a result where these coordinate values are reproduced in the image data 200.

[0042]

<Embodiment 1 : summary>

As discussed thus far, the image processing device 200 according to the embodiment 1 : quantizes coordinate values in the image data 200 ; assigns classifications within the learner 150 to the quantized coordinate values ; and classifies each coordinate value into classifications belonging to XY coordinates of the detection target in each sub region and into classifications not belonging to them. Accordingly, it is possible to handle an object detection problem as a classification problem, thereby improving learning efficiency of the learner 150.

[0043]

In the embodiment 1, the coordinate separator 120 separates coordinate values in the image data 200 for each of coordinate axes. This is for assigning internal classifications of the learner 150 for each of the coordinate axes, thereby simplifying the learning process for each coordinate axis to improve efficiency of the overall learning process. If sufficient efficiency is achieved without separating coordinate values into each axis, it is not always necessary to separate coordinate values.

[0044]

<Embodiment 2>

Figure 6 is a functional block diagram of the image processing device 100 according to an embodiment 2 of the present disclosure. The image processing device 100 according to the embodiment 2 comprises a shifter 180 in addition to the configuration described in the embodiment 1. Other configurations are same as in the embodiment 1.

[0045]

Figure 7 is a diagram showing a process of the shifter 180. If XY coordinates of right eye / left eye / mouth reproduced by the reproducer 170 are shifted from the center of the image data 200, the shifter 180 shifts whole of the image data 200 towards the center. For example, in a case of facial image, the image data 200 is shifted so that the centroid of XY coordinates of right eye / left eye / mouth is positioned at the center of the image data 200, thereby positioning right eye / left eye / mouth in the image data 200 in a balanced manner. The specific shifting process depends on the object in the image data 200. Thus such specific method of shifting the image data 200 may be defined according to the object.

[0046]

<Embodiment 3>

An embodiment 3 of the present disclosure shows an example of definitions of sub regions in a case where the image data 200 is not a facial image. The configurations of the image processing device 100 are same as those described in the embodiments 1-2.

[0047]

Figure 8 is a diagram showing a process of the region definer 110 and a process of the

quantizer 130 when the image data 200 is an image of a running car captured from a side of the car. Since a car runs on a road, it is required to detect the car around the road only. Then the region definer 110 defines a lower half of the image data 200 as a road region, and the quantizer 130 quantizes coordinate values in the road region. Subsequent processes may be performed to the road region only. Then it is possible to improve the processing efficiency without sacrificing the detection accuracy.

[0048]

Figure 9 is a diagram showing a process of the region definer 110 when the image data 200 is an image of a car captured from front of the car. This embodiment assumes that : left and right lights and a number plate are characteristic elements when seeing the car from the front of the car. Thus the region definer 110 defines sub regions corresponding to those elements. Since there could be various car designs, the sub regions in Figure 9 are overlapped with each other.

[0049]

<Embodiment 4>

Figure 10 is a diagram showing hardware configuration of the image processing device 100 according to an embodiment 4 of the present disclosure. Each of the functional units included in the image processing device 100 described in the embodiments 1-3 may be achieved by software implementing those functionalities executed by processors such as CPU (Central Processing Unit).

[0050]

In addition, those functional units may be distributed into a plurality of devices, and each device may communicate with each other, thereby constructing the image processing device 100. In Figure 10, the functionalities of the image processing device 100 are divided into the device 1000a and the device 1000b, and these devices communicate with each other via the network 1400, thereby constructing the image processing device 100. In Figure 10, each of the devices 1000 includes a CPU 1100, a memory 1200, and a communication interface 1300.

[0051]

Any one of the devices 1000 may execute any one of the functionalities. For example : the device 1000a executes processes corresponding to the region definer 110, the coordinate separator 120, the quantizer 130, and the mapper 140, and sends the executed result to the device 1000b ; the device 1000b executes processes corresponding to the learner 150, the classifier 160, the reproducer 170, and the shifter 180.

[0052]

<Embodiment 5>

In the embodiments 1-4, the classifier 160 performs logistic regression analysis to the output from the fully connected layer. Logistic regression analysis is for classifying the output from the fully connected layer. Thus such process may be implemented as a classifying process by neural network. Therefore, instead of performing logistic regression analysis, the classifier 160 may be constructed as a new neural network layer subsequent to the fully connected layer.

[0053]

In the embodiments 1-4, ReLU is used as an activation function of neurons. Other types of activation functions may be used. Examples of such functions include hyperbolic tangent function, sigmoid function, and so on. Each layer may use different activation functions.

[0054]

In the embodiments 1-4, the sub regions and the quantized coordinates are rectangular-based. Other shapes such as triangle or polygon may be used depending on the shape of detected object.

[0055]

The neural network structure of the learner 150 described in the embodiments 1-4 is merely an example, and other structures may be used.

[0056]

<Modifications of the Present Disclosure>

The respective embodiments of the present disclosure have been described above. However, the present disclosure includes various modifications without departing from the

subject matter of the present disclosure. For example, in the abovementioned embodiments, in order to easily understand the present disclosure, the specific configurations are described. However, the present disclosure does not always provide all of the configurations described above. Also, a part of one configuration example can be replaced with another configuration example, and the configuration of one embodiment can be added with the configuration of another embodiment. Also, in a part of the respective configuration examples, another configuration can be added, deleted, or replaced.

[0057]

Also, some or all of the above-described respective configurations, functions, processors, processing means may be realized, for example, as an integrated circuit, or other hardware. Also, the above respective configurations and functions may be realized by allowing the processor to interpret and execute programs for realizing the respective functions. That is, the respective configurations and functions may be realized by software. The information on the program, table, and file for realizing the respective functions can be stored in a storage device such as a memory, a hard disc, or an SSD (Solid State Drive), or a storage medium such as an IC card, an SD card, or a DVD.

List of Reference Signs

[0058]

100 : image processing device

110 : region definer

120 : coordinate separator

130 : quantizer

140 : mapper

150 : learner

160 : classifier

170 : reproducer

180 : shifter

200 : image data

CLAIMS

[Claim 1]

An image processing method for processing image data including an object using a neural network that performs learning to classify input data, comprising :

a quantization step of performing quantization of a pixel coordinate of the image data ;

a mapping step of assigning, to the quantized pixel coordinate, any one of internal classifications internally included in the neural network ;

a learning step of performing learning of the neural network using the quantized pixel coordinate of the image data and using the internal classification assigned in the mapping step ;

an output step of acquiring an output value from the neural network after the learning is performed ;

a reproducing step of performing inverse-quantization of the quantization to the output value from the neural network after the learning is performed, thereby projecting the output value from the neural network onto an coordinate system in the image data to reproduce a position of the object on the coordinate system.

[Claim 2]

The image processing method according to Claim 1, further comprising a step of defining a plurality of regions on the image data,

wherein the image processing method performs the quantization step and the mapping step for each of the regions.

[Claim 3]

The image processing method according to Claim 1, further comprising a step of separating the pixel coordinate into each coordinate axis,

wherein the mapping step assigns different ones of the internal classifications for each of the separated coordinate axes.

[Claim 4]

The image processing method according to Claim 1, further comprising a step of

separating the pixel coordinate into each coordinate axis,

wherein the mapping step assigns different ones of the internal classifications for each of the separated coordinate axes,

the image processing method further comprising a classification step of classifying the output value from the neural network into any one of coordinate values of the object on the coordinate axis,

wherein the reproducing step performs inverse-quantization of the quantization to a result of the classification step, thereby reproducing a position of the object on the coordinate system.

[Claim 5]

The image processing method according to Claim 1, further comprising a step of shifting, towards a center of the image data, the object reproduced by the reproducing step on the coordinate system.

[Claim 6]

An image processing device that processes image data including an object using a neural network that performs learning to classify input data, comprising :

a quantizer that performs quantization of a pixel coordinate of the image data ;

a mapper that assigns, to the quantized pixel coordinate, any one of internal classifications internally included in the neural network ;

a learner that performs learning of the neural network using the quantized pixel coordinate of the image data and using the internal classification assigned by the mapper ;

a reproducer that performs inverse-quantization of the quantization to an output value from the neural network after the learning is performed, thereby projecting the output value from the neural network onto an coordinate system in the image data to reproduce a position of the object on the coordinate system.

[Claim 7]

The image processing device according to Claim 6, further comprising a region definer that defines a plurality of regions on the image data,

wherein the image processing device performs a process of the quantizer and a

process of the mapper for each of the regions.

[Claim 8]

The image processing device according to Claim 6, further comprising a separator that separates the pixel coordinate into each coordinate axis,

wherein the mapper assigns different ones of the internal classifications for each of the separated coordinate axes.

[Claim 9]

The image processing device according to Claim 6, further comprising a separator that separates the pixel coordinate into each coordinate axis,

wherein the mapper assigns different ones of the internal classifications for each of the separated coordinate axes,

the image processing device further comprising a classifier that classifies the output value from the neural network into any one of coordinate values of the object on the coordinate axis,

wherein the reproducer performs inverse-quantization of the quantization to a result of the classifier, thereby reproducing a position of the object on the coordinate system.

[Claim 10]

The image processing device according to Claim 6, further comprising a shifter that shifts, towards a center of the image data, the object reproduced by the reproducer on the coordinate system.