Design of human-like behavior learning machine based on neural network FPGA

Ming Tang, Shengli Hao
Xi’an Peihua University, Xi’an, China

Corresponding author and e-mail: Ming Tang, 81408240@qq.com

Abstract. Neural network has been widely used in image recognition, because it can imitate the behavior of biological visual nerve to obtain high recognition accuracy. Aiming at image recognition, this paper studies the machine self-learning system of image recognition results based on neural network FPGA, designs the data sensing and image recognition processing module based on neural network, and studies the key problems of the system self-recognition and machine self-learning process. The image recognition module uses bionics and parallax principle to get the synchronous exposure image of surrounding scenery, reconstruct the three-dimensional shape and position of surrounding objects, and get the information needed by the system design through image processing algorithm. On the basis of this module, a machine self-learning algorithm based on the result of image recognition is designed to build a knowledge base for the machine and guide the machine by modifying the weight and making decision to realize human-like behavior of machine self-learning. The experimental results show that the recognition accuracy of dynamic image recognition in image recognition network is about 80% compared with the corresponding static image recognition, and the time and power consumption are reduced by 24%, 27.8% and 41% respectively.

1. Introduction

Neural network has been widely used in image recognition, because it can imitate the behavior of biological visual nerve to obtain high recognition accuracy. Recently, the rapid growth of modern applications based on deep learning algorithms has further improved research and implementation. In particular, a variety of neural network-based hardware architecture platforms of FPGA have been proposed, which have the advantages of high performance, reconfigurable, rapid development cycle and so on.

In image recognition, the initial state of the computer can only recognize the basic information of the pixels, which is the same as the vision of the organism. The reason why the organism can distinguish objects is the result of the original image processing by the biological nervous system. The key point of image recognition is that image recognition programming is the comprehensive processing of the original image point information. Image recognition usually includes contour recognition, feature recognition, color recognition, material recognition, object recognition and so on. Generally, the outline of an object can be obtained from the information of color and brightness. According to the data corresponding to the outline, the content of the outline is determined as what object or what feature, and the judgment of the feature and object can not be separated from the processing of the outline and the corresponding logical data. The feature of material recognition is to recognize according to the degree of reflection of the problem. It can not be separated from the...
recognition of contour and the judgment of logical data. Therefore, contour recognition is the most important part in image recognition.

Supervised learning begins before learning begins. A sample training set consisting of several known input vectors and corresponding target variables is provided to the neural network tower to adjust the connection weights between neurons by the difference between the given input value and the expected output value and the actual network output value. Unsupervised learning. This learning method only needs to provide input to the neural network, and does not need the expected output value. The neural network Intensive learning. This algorithm does not need to give a clear expected output, but uses an evaluation mechanism to evaluate the quality factor of the output of the neural network corresponding to a given input. The external environment only gives the evaluation results for the output, and improves the system performance by strengthening the award action.

Reconfigurable FPGA has high expansibility and can be applied in many fields such as automation, intelligent control, computer vision system, artificial intelligence, machine learning and so on.

2. System Architecture Design

2.1. System Design Overall Architecture
The system architecture is shown in Figure1. For each part of the system function flow control, image data capture, processing, storage are used separately:

2.2. Functional Design of Modules in Embedded System Based on FPGA

2.2.1 Data Sensing and Recognition Module
Data perception and recognition module uses binocular camera. Binocular camera uses bionics and parallax principle to get synchronous exposure image of surrounding scenery by calibrated dual cameras. Based on parallax principle, the three-dimensional geometric information of object can be recovered, and the three-dimensional shape and position of surrounding object can be reconstructed. The acquisition of three-dimensional information is based on the principle of triangulation, that is, a triangle is formed between the image plane of two cameras and the object being measured. Through the position relationship between the two cameras, the three-dimensional dimensions of objects in the common field of view of the two cameras and the three-dimensional coordinates of the feature points of space objects can be obtained.
Figure 2. Principle of binocular stereo imaging with horizontal vision

The figure above shows a simple stereo imaging principle of head-up binocular vision. The distance between the projection centers of the two cameras, i.e. the baseline distance B. Two cameras look at the same feature point P of space-time object at the same time, and get the image of point P on "left eye" and "right eye" respectively, their coordinates are respectively \( P_l = (x_l, y_l) \); \( P_r = (x_r, y_r) \).

If the images of two cameras are fixed on the same plane, the Y coordinates of the image coordinates of the feature point P must be the same, that is to say \( y_l = y_r = y \). From the triangular geometric relations, the following relations can be obtained.

\[
\begin{align*}
X_l &= f \frac{x_c}{z_c} \\
X_r &= f \frac{(x_c - B)}{z_c} \\
Y &= f \frac{y_c}{z_c}
\end{align*}
\]

(1)

Then the parallax is as follows \( D = X_l - X_r \). From this, the three-dimensional coordinates of feature point P in camera coordinate system can be calculated:

\[
\begin{align*}
x_c &= \frac{B \cdot X_l}{D} \\
y_c &= \frac{B \cdot Y}{D} \\
z_c &= \frac{B \cdot f}{D}
\end{align*}
\]

(2)

2.2.2 Image Data Processing Module

Image data processing module cameras capture images and output them as analog or digital video signals. The image signals are preprocessed, segmented, extracted and recognized by image processing algorithms (such as filtering, image segmentation, transformation, geometric morphology analysis, etc.). Finally, the information needed for system design is obtained.
2.2.3 Embedded System
The system is designed and implemented based on cloud server scalable FPGA devices, ARM chips and convolution algorithm. The main idea of the system is to divide the neural network algorithm into several basic operations, which can be reconfigured and applied to different projects by compiling different self-learning image processing algorithms. We use FPGA AT3000 series chips (XC3000 series compatible), all memory uses dual-port ARM. Under XILINX development system, the self-learning algorithm circuit is designed with chrysanthemum chain master-slave configuration. After logical simulation, timing simulation, configuration file generation, and then offline experiments.

2.2.4 Self-learning Algorithms
On the basis of open source convolution, reinforcement learning and supervised learning, we add our self-compiled machine self-learning algorithm to build a knowledge base for the machine, and guide the decision-making of the machine by modifying the weights. Finally, the human-like behavior of machine self-learning is realized.

The captured image data is processed locally and sent to the cloud server. After analysis and comparison, some codes are returned:

```python
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
frame_with_text = cv2ImgAddText(frame, tags_str, 0, 0, (255, 255, 255))

if cv2.waitKey(1) == 27:
    break  # esc to quit
```

# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()

---

**Figure 3. Image processing**
3. Experiments and Assessment

3.1. Test environment and process
In this paper, the dynamic reconstruction design of image recognition network, data sensing and recognition module uses binocular camera for image capture. The image data processing module outputs the captured image as analog or digital video signal, preprocesses, segments, extracts and recognizes the image signal through image processing algorithms (such as filtering, image segmentation, transformation, geometric morphology analysis, etc.); finally obtains the information needed by the system design, and then outputs and inspects it in various forms through the external equipment connected with it. Measurements and responses. The code is written in Python description language, the background server is Aliyun server, and the software environment is Windows 10.

The following code is the image data acquisition and transmission part. The captured image data is processed and sent to the cloud server. After analysis and comparison, the results are returned.

```python
def cv2ImgAddText(img, text, left, top, textColor=(0, 255, 0), textSize=20):
    img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    fontText = ImageFont.truetype('font/simsun.ttc', textSize, encoding='utf-8')
    draw = ImageDraw.Draw(img)
    draw.text((left, top), text, textColor, font=fontText)
    return cv2.cvtColor(np.array(img), cv2.COLOR_RGB2BGR)

cap = cv2.VideoCapture(1)
faceCascade = cv2.CascadeClassifier("hearcascade_frontalface_default.xml")
last_detect = 0
tags_str = "aaaa"
while(True):
    # Capture frame-by-frame
    ret, frame = cap.read()
    if not ret:
        continue
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags = cv2.CASCADE_SCALE_IMAGE
    )
    if time.time() - last_detect >= 4:
        last_detect = time.time()
        try:
            enc_ret, img_data = cv2.imencode('.jpg', frame)
            img_base64_data = base64.b64encode(img_data).decode("utf8")
            ret = api.request_data_plus("https://dtplus-cn-shanghai.data.aliyuncs.com",
                                      "type": 1,
                                      "content": img_base64_data
                                      )
            tags = [x['value'] for x in ret['tags']]
```

3.2. Time Power Evaluation
The configuration speed of dynamic reconfiguration of cloud server’s FPGA is basically limited by the size of configuration file and the bandwidth of ICAP port. In the design of this paper, four reconfiguration areas are divided, each reconfiguration area has three time-sharing reconfiguration modules, so a complete image recognition network needs a static configuration file and 12 partial configuration files.

Extensible FPGA development boards, ICAP configuration ports with maximum bandwidth of 3.2 Gb/s at 100 MHz maximum clock frequency. With the recognition time of convolution neural network itself, the total time is 20.3 Ms. Compared with the software implementation based on ARM Cortex A9 processor, as shown in Table 1, there is a great improvement in time and power consumption. Table 1 below shows the time and power consumption of 100 different dynamic and static identifications.
Table 1. Time and Power Consumption of 100 Recognitions for Different Implementation Modes

| platform      | Realization way       | Use time /ms | Power consumption /W |
|---------------|-----------------------|--------------|----------------------|
| ARM Cortex A9 | Software              | 5120         | 100                  |
| Xilinx.VC-707 | Dynamic reconfiguration| 1700         | 76                   |

4. Conclusion
In this paper, for the first time, the dynamic reconfiguration technology of cloud server based on FPGA is applied to the hardware implementation of convolutional neural network, which can effectively solve the problems of low parallel speed, and computing speed cannot meet the real-time requirements of the field, stability, power and size limitations of the chip. It is hoped that in the future, scientific researchers can improve this progress and better promote the development of neural networks and artificial intelligence.

Acknowledgement
This work was supported by Shaanxi Natural Science Fund Project (2018JM7037025).

References
[1] Lu Lu. Study of convolutional neural network and its applications on license plate recognition [D]. Hefei: Hefei University of Technology, 2006.
[2] Shengyu Zhou. Realization of Dynamic Partial Reconfiguration System Based on FPGA [D]. Beijing: Graduate School of Chinese Academy of Sciences, 2007.
[3] Peilong Liu. Research and design of neural network hardware implementation based on FPGA [D]. University of Electronic Science and Technology, 2012.