

《物联网与嵌入式技术》

大作业课题报告

文字-盲文转换器

学 生 姓 名：_____刘昕昀_____

学 科、 专 业：_____人工智能_____

学 号：_____22309236_____

完 成 日 期：_____2024. 04. 29_____

大连理工大学

Dalian University of Technology

目录

1 背景及概述.....	1
2 项目分工与负责任务.....	1
3 图片信息的预处理.....	1
4 图片信息的传输.....	2
5 TCP 心跳校验.....	3
6 总结.....	4
7 致谢.....	4

1 背景及概述

文字盲文转换器是一种用于将普通文字转换成盲文的技术设备。盲文是一种盲人用于阅读和书写的点字系统，通过凹凸不平的点字来表示字母、数字和标点符号。文字盲文转换器的出现，大大方便了盲人阅读普通文字内容的能力，让他们能够更方便地获取信息和知识。

文字盲文转换器通常由软件和硬件组成，软件用于将普通文字转换成盲文的代码，硬件则用于输出实际的盲文点字。文字盲文转换器的应用领域非常广泛，可以用于教育、媒体、出版、图书馆等领域，帮助盲人更好地融入社会和获取信息资源。

2 项目分工与负责任务

项目题目为“文字-盲文转换器”，项目小组为第 10 组，组长为叶书睿，小组成员分别为：魏佳琪、李勇、单文超、李凯、李泽宇、高鹏翔、林源清、刘昕昀、王晶波。本人参与的是文字盲文转换器中图片信息的预处理和传输，首先，利用 OpenCV 对截图进行压缩和转换成 JPEG 格式，并将其编码为字节流。接着，通过 TCP socket 将字节流发送到服务器。在传输过程中可能出现网络问题，例如连接超时，因此需要进行相关处理。为了确保连接稳定，我们会增加心跳校验和连接建立情况的确认，通过标识符标注数据类型，并确认数据接收情况。

3 图片信息的预处理

在 Python 中，使用 OpenCV 对截图进行压缩并转换为 JPEG 格式的基本步骤如下：

- (1) 使用 OpenCV 读取截图；
- (2) 选择一个压缩方法，例如更改图像的比特深度或应用阈值操作；
- (3) 使用 `cv2.imwrite()` 函数将图像保存为 JPEG 格式；
- (4) 将图片编码为字节流。

部分实现代码如图 1 所示。

```

1  import cv2
2  import numpy as np
3
4  # 读取截图
5  image = cv2.imread('screenshot.png')
6
7  # 压缩图片
8  quality = 50 # 压缩质量可以根据需要调整, 范围从0到100
9  encode_params = [int(cv2.IMWRITE_JPEG_QUALITY), quality]
10
11 # 将图片编码为字节流
12 byte_io = cv2.imencode('.jpg', image, encode_params)[1]
13 jpeg_data = byte_io.tobytes()

```

图 1 图片信息预处理

4 图片信息的传输

使用 `socket` 库来通过 TCP socket 发送字节流到服务器，完成图片信息的传输。确保替换 `server_address` 变量中的 IP 地址和端口为我们的服务器信息。`sendall` 方法会尝试发送所有的数据，如果无法发送全部数据，它会引发一个异常。如果需要接收服务器的响应，也可以使用 `recv` 方法实现。

实现图片信息传输的代码如下。

```

import socket

# 服务器的 IP 地址和端口
SERVER_IP = 'IP 地址'
SERVER_PORT = 端口号

# 创建一个 socket 对象
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 连接到服务器
client_socket.connect((SERVER_IP, SERVER_PORT))

# 要发送的字节流数据
message = b'字节流数据'

```

```
# 发送数据到服务器
client_socket.send(message)

# 关闭 socket 连接
client_socket.close()
```

5 TCP 心跳校验

心跳校验的原理是：客户端每隔 N 秒向服务端发送一个心跳消息，服务端收到心跳消息后，回复同样的心跳消息给客户端。如果服务端或客户端在 M 秒（ $M > N$ ）内都没有收到包括心跳消息在内的任何消息，即心跳超时，我们就认为目标 TCP 连接已经断开了。

心跳校验的作用是：（1）通知服务器，客户端存活状态，一旦服务器监测到在某段时间没有收到客户端发来的心跳包，服务器就会释放曾经为此客户端分配的所有资源，例如 Socket 连接；（2）定时的刷新 NAT 内外网 IP 映射表，以便防止 NAT 路由器移除映射表，导致客户端和服务器端的连接中断用户体验。

心跳校验的部分代码实现如图 2 所示。

```
1  import socket
2
3  def setup_tcp_heartbeat(sock):
4      # 设置心跳间隔为30秒
5      sock.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)
6      # 设置心跳的时间间隔和重试间隔（以秒为单位）
7      sock.ioctl(socket.SIO_KEEPALIVE_VALS, (15, 30, 6)) # (On, 15s, 30s)
8
9  def connect_and_confirm(host, port):
10     # 创建一个TCP套接字
11     sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12
13     # 设置TCP心跳
14     setup_tcp_heartbeat(sock)
15
16     # 尝试连接到服务器
17     try:
18         sock.connect((host, port))
19         print("连接已建立。")
20     except socket.error as e:
21         print(f"连接失败: {e}")
```

图 2 心跳校验

6 总结

本文介绍了文字盲文转换器中图片信息的预处理和传输的实现方法和过程,首先,利用 OpenCV 对截图进行压缩和转换成 JPEG 格式,并将其编码为字节流。接着,通过 TCP socket 将字节流发送到服务器。在传输过程中可能出现网络问题,例如连接超时,因此需要进行相关处理。为了确保连接稳定,我们会增加心跳校验和连接建立情况的确认,通过标识符标注数据类型,并确认数据接收情况。最后,成功实现了文字盲文转换器中图片信息的预处理和传输的模块。经过实验,该项目在一定程度上有助于帮助盲人完成阅读。

7 致谢

感谢王开宇老师对《物联网与嵌入式技术》这门课程的付出,在课堂上让我学到了很多实用的知识,使我对于物联网与嵌入式技术有了更深刻的理解,并且在答辩环节老师对于我们的项目提出了非常宝贵的建议,使我们的项目更加完善、具体。同时感谢一同完成这个项目的队友们,是大家的共同努力才有了项目最终的圆满完成。