

1 源代码

1.1 Child_File_Setting.py

```

1  import json
2  import time
3  from PyQt5.QtWidgets import QWidget
4  from PyQt5 import QtWidgets
5  from CvPyGui.ui import child_file
6  Ui_ChildWindow_2 = child_file.Ui_Form
7  class child_file_setting(QWidget,Ui_ChildWindow_2):
8      # 功能：选择文件路径。。显示当前的文件路径。。
9      # 选择音视频保存的格式 是名字的格式还是文件格式？
10     def __init__(self):
11         super().__init__()
12         self.setupUi(self)
13         self.InitChildWindwo()
14     def InitChildWindwo(self):
15         # 初始化,,,读取 json 文件,显示当前的保存路径
16         with open("config/db.json","r",encoding='UTF-8') as
            ↪ dbfile_r:
17             file_save_path = json.load(dbfile_r)
18         self.file_path_label.setText(file_save_path["save_pat
            ↪ h"])
19         self.CreateButtons()
20         if file_save_path["video_format"] == ".avi":
21             self.radioButton_avi.setChecked(True)
22         else:
23             self.radioButton_mp4.setChecked(True)
24         if file_save_path["voice_format"] == ".wav":
25             self.radioButton_wav.setChecked(True)
26         else:
27             self.radioButton_mp3.setChecked(True)
28     def msg(self):
29         m = QtWidgets.QFileDialog.getExistingDirectory(None,
            ↪ "选取文件夹", "C:/") # 起始路径
30         with open("config/db.json","r",encoding='UTF-8') as
            ↪ file_r:
31             savepath = json.load(file_r)

```

```
32     # 原路径保存至 former_save_path
33     former_save_path = savepath["save_path"]
34     print(m) # 打印刚刚获取的当前路径
35     #如果获取的路径为空，就不改变路径，
36     ↪ 把之前的路径赋给新的路径
37     if m == "":
38         m = former_save_path
39     print(m)
40     savepath["save_path"] = m
41     # 将更改后（或获取空时不更改）的路径写入 json 文件
42     with open("config/db.json", "w", encoding='UTF-8') as
43     ↪ dbfile:
44         json.dump(savepath, dbfile)
45     time.sleep(0.5)
46     self.update_file_save_path()
47 def update_file_save_path(self):
48     with open("config/db.json", "r", encoding='UTF-8') as
49     ↪ dbfile_r:
50         file_save_path = json.load(dbfile_r)
51     self.file_path_label.setText(str(file_save_path['save_
52     ↪ _path']))
53 def ok_close(self):
54     self.close()
55 def mp3_button(self):
56     if self.radioButton_mp3.isChecked() == True:
57         self.voiceformat = ".mp3"
58     else:
59         self.voiceformat = ".wav"
60     with open("config/db.json", "r", encoding='UTF-8') as
61     ↪ dbfile:
62         json_file = json.load(dbfile)
63     json_file["voice_format"] = self.voiceformat
64     with open("config/db.json", "w", encoding='UTF-8') as
65     ↪ dbfile:
66         json.dump(json_file, dbfile)
67     print(self.voiceformat)
68 def video_radio_button(self):
69     if self.radioButton_mp4.isChecked() == True:
70         self.videoformat = ".mp4v"
```

```

65         else:
66             self.videoformat = ".avi"
67         with open("config/db.json", "r", encoding='UTF-8') as
        ↪ dbfile:
68             json_file = json.load(dbfile)
69             json_file["video_format"] = self.videoformat
70         with open("config/db.json", "w", encoding='UTF-8') as
        ↪ dbfile:
71             json.dump(json_file, dbfile)
72         print(self.videoformat)
73     def CreateButtons(self):
74         self.choice_file_path_button.clicked.connect(self.msg)
75         self.file_ok.clicked.connect(self.ok_close)
76         self.radioButton_mp3.clicked.connect(self.mp3_button)
77         self.radioButton_wav.clicked.connect(self.mp3_button)
78         self.radioButton_mp4.clicked.connect(self.video_radio_
        ↪ _button)
79         self.radioButton_avi.clicked.connect(self.video_radio_
        ↪ _button)

```

1.2 FilterCvQtContainer.py

```

1  from PyQt5.QtCore import Qt
2  from PyQt5.QtWidgets import (QWidget, QLabel, QHBoxLayout,
3                               QPushButton, QSlider)
4  import cv2
5  import numpy as np
6  class Filter(QWidget):
7       """Common base class for all filters"""
8       defaultK = 3
9       filterCount = 0
10     def __init__(self, name, minValue, maxValue, init,
        ↪ num_of_k, parent=None):
11         super().__init__()
12         self.filter_number = Filter.filterCount
13         self.name = name
14         self.num_of_k = num_of_k
15         self.k = [init]

```

```
16         # Increase the number of filters created
17         Filter.filterCount += 1
18         # Set maximum height
19         self.setMaximumHeight(65)
20         # Variable for the slider/label layout
21         self.lay = QHBoxLayout(self)
22         # Variable for the constant of the OpenCV filter
23         self.k[0] = self.defaultK
24         # Label for the slider
25         self.k_lbl = [QLabel(str(self.k[0]))]
26         # Name for the slider
27         self.name_lbl = QLabel(self.name + ': ')
28         # Set default parameters
29         self.setParameters(minValue, maxValue)
30         # Create delete button
31         self.delete_filter_btn = QPushButton('X')
32         self.delete_filter_btn.clicked.connect(self.deleteFilter)
33         # Adds the slider and it's label to the layout
34         self.createLayout()
35         # Function sending the slider signal to the processing
36         # function
37         self.thresh_sld.valueChanged.connect(self.changeValue)
38     def setParameters(self, minValue, maxValue):
39         # Creates the slider for the OpenCV filter, with min,
40         # max, default and
41         # step values
42         self.thresh_sld = QSlider(Qt.Horizontal, self)
43         self.thresh_sld.setFocusPolicy(Qt.NoFocus)
44         self.thresh_sld.setMinimum(minValue)
45         self.thresh_sld.setMaximum(maxValue)
46         self.thresh_sld.setValue(self.k[0])
47         self.thresh_sld.setSingleStep(2)
48     def createLayout(self):
49         # Adds the slider and its label to the bottom of the
50         # main layout
51         self.lay.addWidget(self.name_lbl)
52         self.lay.addWidget(self.k_lbl[0])
53         self.lay.addWidget(self.thresh_sld)
```

```

51         self.lay.addWidget(self.delete_filter_btn)
52     def changeValue(self, value):
53         # Function for setting the value of k1
54         if value % 2 == 1:
55             self.k[0] = value
56         else:
57             self.k[0] = value + 1
58         self.thresh_sld.setValue(self.k[0])
59         self.k_lbl[0].setText(str(self.k[0]))
60         self.parent().parent().updateImages()
61     def resetValue(self):
62         # Resets the K value to it's default
63         self.changeValue(self.defaultK)
64     def deleteFilter(self):
65         self.parent().parent().deleteFilter(self.filter_numbe
        ↪ r)
66     def process(self, cv_before, name):
67         k = self.k[0]
68         kernel = np.ones((k, k), np.uint8)
69         if name == 'Invert':
70             cv_before = cv2.cvtColor(cv_before,
        ↪ cv2.COLOR_RGB2GRAY)
71             cv_after = cv2.bitwise_not(cv_before)
72         elif name == 'Histogram Equalization':
73             cv_before = cv2.cvtColor(cv_before,
        ↪ cv2.COLOR_RGB2GRAY)
74             clahe = cv2.createCLAHE(clipLimit=2.0,
        ↪ tileGridSize=(8, 8))
75             cv_after = clahe.apply(cv_before)
76         elif name == 'Threshold':
77             cv_before = cv2.cvtColor(cv_before,
        ↪ cv2.COLOR_RGB2GRAY)
78             ret, cv_after = cv2.threshold(
79                 cv_before, k, 255, cv2.THRESH_BINARY)
80         elif name == 'Gaussian Threshold':
81             cv_before = cv2.cvtColor(cv_before,
        ↪ cv2.COLOR_RGB2GRAY)
82             cv_after = cv2.adaptiveThreshold(cv_before, 255,
        ↪ cv2.ADAPTIVE_THRESH_GAUSSIAN_C,

```

```

83                                     cv2.THRESH_BINARY)
84                                     ↪ Y, k,
85                                     ↪ 2)
86
87 elif name == 'HSV':
88     cv_before = cv2.cvtColor(cv_before,
89                               ↪ cv2.COLOR_RGB2HSV)
90     lower_color = np.array([k - 35, 0, 0])
91     upper_color = np.array([k + 35, 255, 255])
92     cv_after = cv2.inRange(cv_before, lower_color,
93                             ↪ upper_color)
94
95 elif name == 'LAB':
96     cv_before = cv2.cvtColor(cv_before,
97                               ↪ cv2.COLOR_RGB2LAB)
98     L, a, b = cv2.split(cv_before)
99     ret, cv_after = cv2.threshold(L, k, 255,
100                                  ↪ cv2.THRESH_BINARY)
101
102 elif name == 'Erosion':
103     cv_before = cv2.cvtColor(cv_before,
104                               ↪ cv2.COLOR_RGB2GRAY)
105     cv_after = cv2.erode(cv_before, kernel,
106                           ↪ iterations=1)
107
108 elif name == 'Dilation':
109     cv_before = cv2.cvtColor(cv_before,
110                               ↪ cv2.COLOR_RGB2GRAY)
111     cv_after = cv2.dilate(cv_before, kernel,
112                           ↪ iterations=1)
113
114 elif name == 'Opening':
115     cv_before = cv2.cvtColor(cv_before,
116                               ↪ cv2.COLOR_RGB2GRAY)
117     cv_after = cv2.morphologyEx(
118         cv_before, cv2.MORPH_OPEN, kernel)
119
120 elif name == 'Closing':
121     cv_before = cv2.cvtColor(cv_before,
122                               ↪ cv2.COLOR_RGB2GRAY)
123     cv_after = cv2.morphologyEx(
124         cv_before, cv2.MORPH_CLOSE, kernel)
125
126 elif name == 'Top Hat':
127     cv_before = cv2.cvtColor(cv_before,
128                               ↪ cv2.COLOR_RGB2GRAY)

```

```

109         cv_after = cv2.morphologyEx(
110             cv_before, cv2.MORPH_TOPHAT, kernel)
111     elif name == 'Black Hat':
112         cv_before = cv2.cvtColor(cv_before,
113             ↪ cv2.COLOR_RGB2GRAY)
114         cv_after = cv2.morphologyEx(
115             cv_before, cv2.MORPH_BLACKHAT, kernel)
116     elif name == 'Canny':
117         cv_before = cv2.cvtColor(cv_before,
118             ↪ cv2.COLOR_RGB2GRAY)
119         cv_after = cv2.Canny(cv_before, 100, k)
120     elif name == 'Laplacian':
121         cv_before = cv2.cvtColor(cv_before,
122             ↪ cv2.COLOR_RGB2GRAY)
123         cv_after = cv2.Laplacian(cv_before, cv2.CV_64F)
124         cv_after = np.absolute(cv_after)
125         cv_after = np.uint8(cv_after)
126     return cv_after

```

1.3 ImageCvQtContainer.py

```

1  from PyQt5.QtGui import *
2  from PyQt5.QtWidgets import *
3  from PyQt5.QtCore import Qt
4  class Image(QWidget):
5      """Common base for the images"""
6      def __init__(self, name, label):
7          super().__init__()
8          self.frame_lbl = label
9          print (self.frame_lbl.size())
10     def updateImage(self, opencv_rgb_image):
11         self.cv_img_rgb = opencv_rgb_image
12         height, width, channel = self.cv_img_rgb.shape
13         bytesPerLine = 3 * width
14         self.q_image = QImage(self.cv_img_rgb.data,
15             ↪ width,height, bytesPerLine, QImage.Format_RGB888)

```

```

15         self.frame_lbl.setPixmap(QPixmap.fromImage(self.q_image).scaled(self.frame_lbl.size(),
    ↳ ge).scaled(self.frame_lbl.size(),
    ↳ aspectRatioMode=Qt.KeepAspectRatio))
16     def saveImage(self):
17         filter = "Images (*.png *.jpg)"
18         image_path, _ = QFileDialog.getSaveFileName(self,
    ↳ filter=filter)
19         cv_img_bgr = cv2.cvtColor(
20             self.cv_img_rgb, cv2.COLOR_RGB2BGR)
21         cv2.imwrite(image_path, cv_img_bgr)

```

1.4 __init__.py

```

1

```

1.5 child_CameraVoice_setting.py

```

1  import json
2  import time
3  import numpy as np
4  import pyaudio
5  from PyQt5.QtCore import (QTimer, pyqtSignal, QThread)
6  from CvPyGui import ImageCvQtContainer
7  import pygame
8  import cv2
9  import threading
10 from PyQt5.QtWidgets import QWidget
11 from CvPyGui.ui import child_test
12 Ui_ChildWindow = child_test.Ui_Form
13 hd_width = 1920
14 hd_highth= 1080
15 place_choice_camera = '请选择摄像头'
16 no_camera_ = '不选择摄像头'
17 place_choice_voice = '请选择麦克风'
18 no_voice = '不选择麦克风'
19 img_no_camera = cv2.imread('config/image/no_camera.jpg',
    ↳ cv2.IMREAD_COLOR)

```



```

20 img_no_voice = cv2.imread('config/image/no_voice.jpg',
    ↪ cv2.IMREAD_COLOR)
21 img_voice =
    ↪ cv2.imread('config/image/voice.jpg', cv2.IMREAD_COLOR)
22 tab1_judge = False # 判断下拉列表选择的内容是否是摄像头
23 tab2_judge = False
24 tab3_judge = False
25 tab4_judge = False
26 camera_judge = {}
27 thread_judge = {}
28 frame = {}
29 frame2 = {}
30 frame3 = {}
31 cap = {}
32 cap2 = {}
33 cap3 = {}
34 tab1_text = ''
35 tab2_text = ''
36 tab3_text = ''
37 tab4_text = ''
38 class UpdateVolume(QThread):
39     update_data = pyqtSignal(str)
40     def __init__(self):
41         super().__init__()
42         self.vo_judge = True
43         self.voice_index_thread = 0
44     def run(self):
45         CHUNK = 1024
46         FORMAT = pyaudio.paInt16
47         CHANNELS = 1
48         RATE = 44100
49         INTERVAL = 5
50         pa = pyaudio.PyAudio()
51         stream = pa.open(format=FORMAT,
52                             channels=CHANNELS,
53                             rate=RATE,
54                             input=True,
55                             frames_per_buffer=CHUNK,

```

```

56         input_device_index=self.voice_index_1
           ↪ thread)

57     buffer = []
58     while self.vo_judge:
59         for i in range(int(INTERVAL * RATE / CHUNK)): #
           ↪ STREAM INTERVAL
60             if self.vo_judge == False:
61                 break
62             data = np.fromstring(stream.read(CHUNK),
           ↪ dtype=np.int16)
63             self.un = int(np.amax(data))
64             self.un = int(pow(self.un,0.5))
65             if self.un >= 100:
66                 self.un = 99
67             self.update_data.emit(str(self.un))
68     print('子线程结束')
69 #1:下拉列表选择后lable标签显示对应的视频。 获取下拉列表的内容
           ↪ 根据内容确定显示的摄像头。
70 class Child_Window(QWidget,Ui_ChildWindow):
71     def __init__(self):
72         super().__init__()
73         Ui_ChildWindow.__init__(self)
74         self.setupUi(self)
75         self.InitChildWindow()
76         self.timer = QTimer()
77         self.child_wind_judge = True
78     def InitChildWindow(self):# 初始化子窗口
79         pygame.init()
80         pygame.camera.init()
81         global cameralist_child
82         cameralist_child = pygame.camera.list_cameras()
83         self.update_tab1_image =
           ↪ ImageCvQtContainer.Image('tab1_label_camera',
           ↪ self.tab1_label)
84         self.update_tab2_image =
           ↪ ImageCvQtContainer.Image('tab2_label_camera',
           ↪ self.tab1_label_3)

```

```

85         self.update_tab3_image =
            ↳ ImageCvQtContainer.Image('tab3_label_camera',
            ↳ self.tab1_label_4)
86     self.update_tab1_image.updateImage(img_no_camera)
87     self.update_tab2_image.updateImage(img_no_camera)
88     self.update_tab3_image.updateImage(img_no_camera)
89     self.tab1_combobox.addItem(pleace_choice_camera)
90     self.tab1_combobox_3.addItem(pleace_choice_camera)
91     self.tab1_combobox_4.addItem(pleace_choice_camera)
92     self.tab1_combobox_7.addItem(pleace_choice_voice)
93     self.tab1_combobox.addItem(no_camera_)
94     self.tab1_combobox_3.addItem(no_camera_)
95     self.tab1_combobox_4.addItem(no_camera_)
96     self.tab1_combobox_7.addItem(no_voice)
97     self.tab1_combobox.addItems(cameralist_child)
98     self.tab1_combobox_3.addItems(cameralist_child)
99     self.tab1_combobox_4.addItems(cameralist_child)
100    self.CreateButtons()
101    with open("config/db.json", "r", encoding='UTF-8') as
        ↳ dbfile_r:
102        camera_voice_name = json.load(dbfile_r)
103        self.tab1_text = camera_voice_name["hd_camera_name"]
104        self.tab2_text = camera_voice_name["face_camera_name"]
105        self.tab3_text = camera_voice_name["eye_camera_name"]
106        self.tab4_text = camera_voice_name["hk_voice_name"]
107        if len(cameralist_child) >= 1:
108            for i in range(len(cameralist_child)):
109                camera_judge[i] = 'close'
110        else:
111            print('no camera')
112    p = pyaudio.PyAudio()
113    info = p.get_host_api_info_by_index(0)
114    numdevices = info.get('deviceCount')
115    global voice_list
116    voice_list = []
117    for i in range(0, numdevices):
118        if (p.get_device_info_by_host_api_device_index(0,
            ↳ i).get('maxInputChannels')) > 0:

```

```

119         voice_list.append(p.get_device_info_by_host_a
↪         pi_device_index(0,
↪         i).get('name'))
120     self.tab1_combobox_7.addItem(voice_list)
121     self.child_wind_judge = True
122     if self.tab1_text in cameralist_child:
123         self.tab1_combobox.setCurrentIndex(cameralist_chi
↪         ld.index(self.tab1_text) +
↪         2)
124     if self.tab2_text in cameralist_child:
125         self.tab1_combobox_3.setCurrentIndex(cameralist_c
↪         hild.index(self.tab2_text) +
↪         2)
126     if self.tab3_text in cameralist_child:
127         self.tab1_combobox_4.setCurrentIndex(cameralist_c
↪         hild.index(self.tab3_text) +
↪         2)
128     if self.tab4_text in voice_list:
129         self.tab1_combobox_7.setCurrentIndex(voice_list.i
↪         ndex(self.tab4_text) +
↪         2)
130     def tab1_combobox_setting(self): #
↪     tab1中的下拉列表选择摄像头
131     # 获取当前下拉列表选中的文本
132     tab1_camera_text = self.tab1_combobox.currentText()
133     self.tab1_text = tab1_camera_text
134     #尝试查询当前文本所在摄像头列表中第几项,
↪     查询不到则显示没有摄像头的图片。
135     try:
136         tab1_camera_index =
↪         cameralist_child.index(tab1_camera_text)
137         tab1_judge = True
138     except:
139         self.update_tab1_image.updateImage(img_no_camera)
140         tab1_judge = False
141     # 获取摄像头的名字后显示画面
142     # 启动一个线程,显示摄像头捕获的画面
143     #如果查询到文本所在该列表中,并且该摄像头没有打开,
↪     就启动线程显示该摄像头的画面

```

```

144         if tab1_judge:
145             if camera_judge[tab1_camera_index] == 'close':
146                 camera_judge[tab1_camera_index] = 'open'
147                 threading._start_new_thread(self.open_camera,
148                                             ↪ (tab1_camera_index,))
149
150             else:
151                 self.update_tab1_image.updateImage(img_no_camera)
152
153     def tab2_combobox_setting(self):
154         tab2_camera_text = self.tab1_combobox_3.currentText()
155         self.tab2_text = tab2_camera_text
156         try:
157             tab2_camera_index =
158                 ↪ cameralist_child.index(tab2_camera_text)
159             tab2_judge = True
160         except:
161             self.update_tab2_image.updateImage(img_no_camera)
162             tab2_judge = False
163         # 获取摄像头的名字后显示画面
164         # 启动一个线程，显示摄像头捕获的画面
165         if tab2_judge:
166             # 如果选择的摄像头没有打开
167             if camera_judge[tab2_camera_index] == 'close':
168                 camera_judge[tab2_camera_index] = 'open'
169                 threading._start_new_thread(self.open_camera,
170                                             ↪ 2,
171                                             ↪ (tab2_camera_index,))
172
173             else:
174                 self.update_tab2_image.updateImage(img_no_camera)
175
176     def open_camera(self, camera_index):
177         #传入摄像头的序号，
178         ↪ 判断摄像头是否开启的代码应该在调用该函数的函数中。
179         #根据摄像头的序号，打开摄像头，
180         ↪ 然后把摄像头的画面传入其他函数？
181         #只有三个要显示的标签，判断要在那个标签上显示，
182         ↪ 然后更新图片。
183         # cap[camera_index] =
184         ↪ cv2.VideoCapture(camera_index, cv2.CAP_DSHOW)
185         cap[camera_index] = cv2.VideoCapture(camera_index)
186         print(cap[camera_index].get(3))

```

```

175         camera_judge[camera_index] = 'open'
176         cap[camera_index].set(3,hd_width)
177         cap[camera_index].set(4,hd_width)
178         cap[camera_index].set(5,30)
179         while self.child_wind_judge:
180             self.ret, frame[camera_index] =
181                 ↪ cap[camera_index].read()
182             if self.ret:
183                 #判断需要在哪几个标签上显示 判断方式:
184                 ↪ 获取三个下拉列表当前显示的值,
185                 ↪ 确定该值在摄像头列表中所占的序号,
186                 ↪ 序号与 camera_index 相同则更新图片。
187                 frame[camera_index] =
188                 ↪ cv2.cvtColor(frame[camera_index],
189                 ↪ cv2.COLOR_BGR2RGB)
190             if (self.tab1_combobox.currentText() !=
191                 ↪ place_choice_camera) and
192                 ↪ (self.tab1_combobox.currentText() !=
193                 ↪ no_camera_):
194                 if camera_index == cameralist_child.index_1
195                 ↪ (self.tab1_combobox.currentText()):
196                     self.update_tab1_image.updateImage(fr_
197                     ↪ ame[camera_index])
198                 else:
199                     pass
200             else:
201                 pass
202             if (self.tab1_combobox_3.currentText() !=
203                 ↪ place_choice_camera) and
204                 ↪ (self.tab1_combobox_3.currentText() !=
205                 ↪ no_camera_):
206                 if camera_index == cameralist_child.index_2
207                 ↪ (self.tab1_combobox_3.currentText()):
208                     self.update_tab2_image.updateImage(fr_
209                     ↪ ame[camera_index])
210                 else:
211                     pass
212             else:
213                 pass
214             pass

```

```

198         if (self.tab1_combobox_4.currentText() !=
        ↪ place_choice_camera) and
        ↪ (self.tab1_combobox_4.currentText() !=
        ↪ no_camera_):
199             if camera_index == cameralist_child.index_
        ↪ (self.tab1_combobox_4.currentText()):
200                 self.update_tab3_image.updateImage(fr_
        ↪ ame[camera_index])
201             else:
202                 pass
203         else:
204             pass
205         if (cameralist_child[camera_index] !=
        ↪ self.tab1_combobox.currentText()) &
        ↪ (cameralist_child[camera_index] !=
        ↪ self.tab1_combobox_3.currentText()) &
        ↪ (cameralist_child[camera_index] !=
        ↪ self.tab1_combobox_4.currentText()):
206             camera_judge[camera_index] = 'close'
207             break
208         if self.child_wind_judge == False:
        ↪ #子窗口如果关闭就停止循环
209             break
210     def open_camera_2(self, camera_index):
211         # cap2[camera_index] =
        ↪ cv2.VideoCapture(camera_index, cv2.CAP_DSHOW)
212         cap2[camera_index] = cv2.VideoCapture(camera_index)
213         camera_judge[camera_index] = 'open'
214         cap2[camera_index].set(3, hd_width)
215         cap2[camera_index].set(4, hd_width)
216         cap2[camera_index].set(5, 30.0)
217         while self.child_wind_judge:
218             self.ret_2, frame2[camera_index] =
        ↪ cap2[camera_index].read()
219             if self.ret_2:
220                 frame2[camera_index] =
        ↪ cv2.cvtColor(frame2[camera_index],
        ↪ cv2.COLOR_BGR2RGB)

```

```

221         if (self.tab1_combobox.currentText() !=
↪         place_choice_camera) and
↪         (self.tab1_combobox.currentText() !=
↪         no_camera_):
222             if camera_index == cameralist_child.index_1
↪             (self.tab1_combobox.currentText()):
223                 self.update_tab1_image.updateImage(fr_1
↪                 ame2[camera_index])
224             else:
225                 pass
226         else:
227             pass
228         if (self.tab1_combobox_3.currentText() !=
↪         place_choice_camera) and
↪         (self.tab1_combobox_3.currentText() !=
↪         no_camera_):
229             if camera_index == cameralist_child.index_1
↪             (self.tab1_combobox_3.currentText()):
230                 self.update_tab2_image.updateImage(fr_1
↪                 ame2[camera_index])
231             else:
232                 pass
233         else:
234             pass
235         if (self.tab1_combobox_4.currentText() !=
↪         place_choice_camera) and
↪         (self.tab1_combobox_4.currentText() !=
↪         no_camera_):
236             if camera_index == cameralist_child.index_1
↪             (self.tab1_combobox_4.currentText()):
237                 self.update_tab3_image.updateImage(fr_1
↪                 ame2[camera_index])
238             else:
239                 pass
240         else:
241             pass

```



```

242         if (cameralist_child[camera_index] !=
↪ self.tab1_combobox.currentText()) &
↪ (cameralist_child[camera_index] !=
↪ self.tab1_combobox_3.currentText()) &
↪ (cameralist_child[camera_index] !=
↪ self.tab1_combobox_4.currentText()):
243             camera_judge[camera_index] = 'close'
244             break
245         else:
246             pass
247         if self.child_wind_judge == False:
↪ #子窗口如果关闭就停止循环
248             break
249     def open_camera_3(self, camera_index):
250         # cap3[camera_index] =
↪ cv2.VideoCapture(camera_index, cv2.CAP_DSHOW)
251         cap3[camera_index] = cv2.VideoCapture(camera_index)
252         camera_judge[camera_index] = 'open'
253         cap3[camera_index].set(3, hd_width)
254         cap3[camera_index].set(4, hd_width)
255         cap3[camera_index].set(5, 30.0)
256         while self.child_wind_judge:
257             self.ret_3, frame3[camera_index] =
↪ cap3[camera_index].read()
258             if self.ret_3:
259                 frame3[camera_index] =
↪ cv2.cvtColor(frame3[camera_index],
↪ cv2.COLOR_BGR2RGB)
260                 if (self.tab1_combobox.currentText() !=
↪ place_choice_camera) and
↪ (self.tab1_combobox.currentText() !=
↪ no_camera):
261                     if camera_index == cameralist_child.index_
↪ (self.tab1_combobox.currentText()):
262                         self.update_tab1_image.updateImage(fr_
↪ ame3[camera_index])
263                 else:
264                     pass
265             else:

```

```

266         pass
267     if (self.tab1_combobox_3.currentText() !=
        ↪ place_choice_camera) and
        ↪ (self.tab1_combobox_3.currentText() !=
        ↪ no_camera_):
268         if camera_index == cameralist_child.index_
        ↪ (self.tab1_combobox_3.currentText()):
269             self.update_tab2_image.updateImage(fr_
        ↪ ame3[camera_index])
270         else:
271             pass
272     else:
273         pass
274     if (self.tab1_combobox_4.currentText() !=
        ↪ place_choice_camera) and
        ↪ (self.tab1_combobox_4.currentText() !=
        ↪ no_camera_):
275         if camera_index == cameralist_child.index_
        ↪ (self.tab1_combobox_4.currentText()):
276             self.update_tab3_image.updateImage(fr_
        ↪ ame3[camera_index])
277         else:
278             pass
279     else:
280         pass
281     if (cameralist_child[camera_index] !=
        ↪ self.tab1_combobox.currentText()) &
        ↪ (cameralist_child[camera_index] !=
        ↪ self.tab1_combobox_3.currentText()) &
        ↪ (cameralist_child[camera_index] !=
        ↪ self.tab1_combobox_4.currentText()):
282         camera_judge[camera_index] = 'close'
283         break
284     else:
285         pass
286     if self.child_wind_judge == False:
        ↪ #子窗口如果关闭就停止循环
287         break
288     def tab3_combobox_setting(self):

```

```

289         tab3_camera_text = self.tab1_combobox_4.currentText()
290         self.tab3_text = tab3_camera_text
291         try:
292             tab3_camera_index =
                ↪ cameralist_child.index(tab3_camera_text)
293             tab3_judge = True
294         except:
295             self.update_tab3_image.updateImage(img_no_camera)
296             tab3_judge = False
297         # 获取摄像头的名字后显示画面
298         # 启动一个线程，显示摄像头捕获的画面
299         if tab3_judge:
300             if camera_judge[tab3_camera_index] == 'close':
301                 camera_judge[tab3_camera_index] = 'open'
302                 threading._start_new_thread(self.open_camera_
                ↪ 3,
                ↪ (tab3_camera_index,))
303         else:
304             self.update_tab3_image.updateImage(img_no_camera)
305     def tab4_voice_setting(self):
306         tab4_voice_text = self.tab1_combobox_7.currentText()
307         self.tab4_text = tab4_voice_text
308         try:
309             voice_index = voice_list.index(self.tab4_text)
310             tab4_judge = True
311         except:
312             tab4_judge = False
313         try:
314             self.sub_thread.vo_judge = False
315         except:
316             pass
317         time.sleep(0.2)
318         if tab4_judge:
319             self.sub_thread = UpdateVolume()
320             self.sub_thread.vo_judge = True
321             self.sub_thread.voice_index_thread = voice_index
322             self.sub_thread.update_data.connect(self.unnn)
323             self.sub_thread.start()
324         else:

```

```

325         print('无声音设备')
326     print('voice setting')
327     def unnn(self,data):
328         self.progressBar.setValue(int(data))
329     def save_camera_json(self):
330         # 用来关闭上面所有的线程, 改写 json 配置文件。
331         # tab1_text、tab2_text、tab3_text分别是行为、面部、
332         #   ↳ 眼部的摄像头名字, 初始内容为读取json文件得到的。
333         with open("config/db.json","r",encoding='UTF-8') as
334         #   ↳ dbfile_r:
335             camera_voice_name = json.load(dbfile_r)
336             camera_voice_name["hd_camera_name"] = self.tab1_text
337             camera_voice_name["face_camera_name"] = self.tab2_text
338             camera_voice_name["eye_camera_name"] = self.tab3_text
339             camera_voice_name["hk_voice_name"] = self.tab4_text
340             with open("config/db.json","w",encoding='UTF-8') as
341             #   ↳ dbfile:
342                 json.dump(camera_voice_name,dbfile)
343             self.child_wind_judge = False
344             try:
345                 self.sub_thread.vo_judge = False
346                 print('success')
347             except:
348                 print('default')
349             self.close()# 关闭该窗口
350     def closeEvent(self, event):
351         print('关闭窗口')
352         self.child_wind_judge = False
353         try:
354             self.sub_thread.vo_judge = False
355             self.child_wind_judge = False
356         except:
357             pass
358     #应该是改变摄像头选项时修改配置文件还是在点确认是关闭呢?
359     #   ↳ 点确认时关闭, 先把选好的信息存起来, 点确认时修改。
360     def CreateButtons(self):
361         # print("createbuttons")
362         self.tab1_combobox.currentIndexChanged.connect(self.t_
363         #   ↳ ab1_combobox_setting)

```

```

359         self.tab1_combobox_3.currentIndexChanged.connect(self
        ↪ .tab2_combobox_setting)
360         self.tab1_combobox_4.currentIndexChanged.connect(self
        ↪ .tab3_combobox_setting)
361         self.tab1_combobox_7.currentIndexChanged.connect(self
        ↪ .tab4_voice_setting)
362         self.OK.clicked.connect(self.save_camera_json)

```

1.6 child_file.py

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file
4  ↪ 'child_file.ui'
5  #
6  # Created by: PyQt5 UI code generator 5.15.10
7  #
8  # WARNING: Any manual changes made to this file will be lost
9  ↪ when pyuic5 is
10 # run again. Do not edit this file unless you know what you
11 ↪ are doing.
12
13
14 from PyQt5 import QtCore, QtGui, QtWidgets
15
16 class Ui_Form(object):
17     def setupUi(self, Form):
18         Form.setObjectName("Form")
19         Form.resize(385, 269)
20         self.file_ok = QtWidgets.QPushButton(Form)
21         self.file_ok.setGeometry(QtCore.QRect(290, 230, 75,
22         ↪ 23))
23         self.file_ok.setObjectName("file_ok")
24         self.label = QtWidgets.QLabel(Form)
25         self.label.setGeometry(QtCore.QRect(10, 100, 101, 16))
26         self.label.setObjectName("label")
27         self.layoutWidget = QtWidgets.QWidget(Form)

```

```
25     self.layoutWidget.setGeometry(QRect(14, 5, 361,
    ↪ 71))
26     self.layoutWidget.setObjectName("layoutWidget")
27     self.verticalLayout =
    ↪ QtWidgets.QVBoxLayout(self.layoutWidget)
28     self.verticalLayout.setContentsMargins(0, 0, 0, 0)
29     self.verticalLayout.setObjectName("verticalLayout")
30     self.horizontalLayout_2 = QtWidgets.QHBoxLayout()
31     self.horizontalLayout_2.setObjectName("horizontalLayo_
    ↪ ut_2")
32     self.show_file_path_label =
    ↪ QtWidgets.QLabel(self.layoutWidget)
33     self.show_file_path_label.setStyleSheet("")
34     self.show_file_path_label.setObjectName("show_file_pa_
    ↪ th_label")
35     self.horizontalLayout_2.addWidget(self.show_file_path_
    ↪ _label)
36     spacerItem = QtWidgets.QSpacerItem(40, 20,
    ↪ QtWidgets.QSizePolicy.Minimum,
    ↪ QtWidgets.QSizePolicy.Expanding)
37     self.horizontalLayout_2.addItem(spacerItem)
38     self.verticalLayout.addLayout(self.horizontalLayout_2)
39     self.horizontalLayout = QtWidgets.QHBoxLayout()
40     self.horizontalLayout.setObjectName("horizontalLayout_
    ↪ ")
41     self.file_path_label =
    ↪ QtWidgets.QLabel(self.layoutWidget)
42     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
    ↪ ICY.Minimum,
    ↪ QtWidgets.QSizePolicy.Preferred)
43     sizePolicy.setHorizontalStretch(0)
44     sizePolicy.setVerticalStretch(0)
45     sizePolicy.setHeightForWidth(self.file_path_label.siz_
    ↪ ePolicy().hasHeightForWidth())
46     self.file_path_label.setSizePolicy(sizePolicy)
47     self.file_path_label.setMinimumSize(QtCore.QSize(0,
    ↪ 20))
48     self.file_path_label.setObjectName("file_path_label")
49     self.horizontalLayout.addWidget(self.file_path_label)
```

```

50     self.choice_file_path_button =
        ↳ QtWidgets.QPushButton(self.layoutWidget)
51     sizePolicy =
        ↳ QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Fixed,
        ↳ QtWidgets.QSizePolicy.Fixed)
52     sizePolicy.setHorizontalStretch(0)
53     sizePolicy.setVerticalStretch(0)
54     sizePolicy.setHeightForWidth(self.choice_file_path_bu
        ↳ tton.sizePolicy().hasHeightForWidth())
55     self.choice_file_path_button.setSizePolicy(sizePolicy)
56     self.choice_file_path_button.setObjectName("choice_fi
        ↳ le_path_button")
57     self.horizontalLayout.addWidget(self.choice_file_path
        ↳ _button)
58     self.verticalLayout.addLayout(self.horizontalLayout)
59     self.layoutWidget1 = QtWidgets.QWidget(Form)
60     self.layoutWidget1.setGeometry(QtCore.QRect(40, 180,
        ↳ 291, 22))
61     self.layoutWidget1.setObjectName("layoutWidget1")
62     self.horizontalLayout_3 =
        ↳ QtWidgets.QHBoxLayout(self.layoutWidget1)
63     self.horizontalLayout_3.setContentsMargins(0, 0, 0, 0)
64     self.horizontalLayout_3.setObjectName("horizontalLayo
        ↳ ut_3")
65     self.label_3 = QtWidgets.QLabel(self.layoutWidget1)
66     self.label_3.setObjectName("label_3")
67     self.horizontalLayout_3.addWidget(self.label_3)
68     self.radioButton_mp3 =
        ↳ QtWidgets.QRadioButton(self.layoutWidget1)
69     self.radioButton_mp3.setEnabled(False)
70     self.radioButton_mp3.setObjectName("radioButton_mp3")
71     self.horizontalLayout_3.addWidget(self.radioButton_mp
        ↳ 3)
72     self.radioButton_wav =
        ↳ QtWidgets.QRadioButton(self.layoutWidget1)
73     self.radioButton_wav.setEnabled(False)
74     self.radioButton_wav.setObjectName("radioButton_wav")
75     self.horizontalLayout_3.addWidget(self.radioButton_wa
        ↳ v)

```

```

76         self.layoutWidget2 = QtWidgets.QWidget(Form)
77         self.layoutWidget2.setGeometry(QtCore.QRect(40, 130,
78             ↪ 291, 22))
79         self.layoutWidget2.setObjectName("layoutWidget2")
80         self.horizontalLayout_4 =
81             ↪ QtWidgets.QHBoxLayout(self.layoutWidget2)
82         self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
83         self.horizontalLayout_4.setObjectName("horizontalLayo_
84             ↪ ut_4")
85         self.label_2 = QtWidgets.QLabel(self.layoutWidget2)
86         self.label_2.setObjectName("label_2")
87         self.horizontalLayout_4.addWidget(self.label_2)
88         self.radioButton_mp4 =
89             ↪ QtWidgets.QRadioButton(self.layoutWidget2)
90         self.radioButton_mp4.setObjectName("radioButton_mp4")
91         self.horizontalLayout_4.addWidget(self.radioButton_mp
92             ↪ 4)
93         self.radioButton_avi =
94             ↪ QtWidgets.QRadioButton(self.layoutWidget2)
95         self.radioButton_avi.setObjectName("radioButton_avi")
96         self.horizontalLayout_4.addWidget(self.radioButton_av
97             ↪ i)
98
99         self.retranslateUi(Form)
100         QtCore.QMetaObject.connectSlotsByName(Form)
101
102     def retranslateUi(self, Form):
103         _translate = QtCore.QCoreApplication.translate
104         Form.setWindowTitle(_translate("Form", " 初始化配置"))
105         self.file_ok.setText(_translate("Form", " 确认"))
106         self.label.setText(_translate("Form",
107             ↪ "音视频格式选择"))
108         self.show_file_path_label.setText(_translate("Form",
109             ↪ "当前文件保存路径:"))
110         self.file_path_label.setText(_translate("Form", "C:"))
111         self.choice_file_path_button.setText(_translate("Form",
112             ↪ "选择文件保存路径"))
113         self.label_3.setText(_translate("Form", " 音频格式"))

```



```

104         self.radioButton_mp3.setText(_translate("Form",
            ↪ "mp3"))
105         self.radioButton_wav.setText(_translate("Form",
            ↪ "wav"))
106         self.label_2.setText(_translate("Form", " 视频格式"))
107         self.radioButton_mp4.setText(_translate("Form",
            ↪ "mp4"))
108         self.radioButton_avi.setText(_translate("Form",
            ↪ "avi"))
109
110
111 if __name__ == "__main__":
112     import sys
113     app = QtWidgets.QApplication(sys.argv)
114     Form = QtWidgets.QWidget()
115     ui = Ui_Form()
116     ui.setupUi(Form)
117     Form.show()
118     sys.exit(app.exec_())

```

1.7 child_file.ui

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3     <class>Form</class>
4     <widget class="QWidget" name="Form">
5         <property name="geometry">
6             <rect>
7                 <x>0</x>
8                 <y>0</y>
9                 <width>385</width>
10                <height>269</height>
11            </rect>
12        </property>
13        <property name="windowTitle">
14            <string>初始化配置</string>
15        </property>
16        <widget class="QPushButton" name="file_ok">

```

```
17     <property name="geometry">
18         <rect>
19             <x>290</x>
20             <y>230</y>
21             <width>75</width>
22             <height>23</height>
23         </rect>
24     </property>
25     <property name="text">
26         <string>确认</string>
27     </property>
28 </widget>
29 <widget class="QLabel" name="label">
30     <property name="geometry">
31         <rect>
32             <x>10</x>
33             <y>100</y>
34             <width>101</width>
35             <height>16</height>
36         </rect>
37     </property>
38     <property name="text">
39         <string>音视频格式选择</string>
40     </property>
41 </widget>
42 <widget class="QWidget" name="layoutWidget">
43     <property name="geometry">
44         <rect>
45             <x>14</x>
46             <y>5</y>
47             <width>361</width>
48             <height>71</height>
49         </rect>
50     </property>
51     <layout class="QVBoxLayout" name="verticalLayout">
52         <item>
53             <layout class="QHBoxLayout" name="horizontalLayout_2">
54                 <item>
55                     <widget class="QLabel" name="show_file_path_label">
```

```
56         <property name="styleSheet">
57             <string notr="true"/>
58         </property>
59         <property name="text">
60             <string>当前文件保存路径:</string>
61         </property>
62     </widget>
63 </item>
64 <item>
65     <spacer name="horizontalSpacer">
66         <property name="orientation">
67             <enum>Qt::Orientation::Horizontal</enum>
68         </property>
69         <property name="sizeHint" stdset="0">
70             <size>
71                 <width>40</width>
72                 <height>20</height>
73             </size>
74         </property>
75     </spacer>
76 </item>
77 </layout>
78 </item>
79 <item>
80     <layout class="QHBoxLayout" name="horizontalLayout">
81         <item>
82             <widget class="QLabel" name="file_path_label">
83                 <property name="sizePolicy">
84                     <sizepolicy hsize="Minimum"
85                        ↪ vsizetype="Preferred">
86                         <horstretch>0</horstretch>
87                         <verstretch>0</verstretch>
88                     </sizepolicy>
89                 </property>
90                 <property name="minimumSize">
91                     <size>
92                         <width>0</width>
93                         <height>20</height>
94                     </size>
```

```

94         </property>
95         <property name="text">
96             <string>C:</string>
97         </property>
98     </widget>
99 </item>
100 <item>
101     <widget class="QPushButton"
102         ↪ name="choice_file_path_button">
103         <property name="sizePolicy">
104             <sizepolicy hsize="Fixed" vsize="Fixed">
105                 <horstretch>0</horstretch>
106                 <verstretch>0</verstretch>
107             </sizepolicy>
108         </property>
109         <property name="text">
110             <string>选择文件保存路径</string>
111         </property>
112     </widget>
113 </item>
114 </layout>
115 </item>
116 </layout>
117 </widget>
118 <widget class="QWidget" name="layoutWidget">
119     <property name="geometry">
120         <rect>
121             <x>40</x>
122             <y>180</y>
123             <width>291</width>
124             <height>22</height>
125         </rect>
126     </property>
127     <layout class="QHBoxLayout" name="horizontalLayout_3">
128         <item>
129             <widget class="QLabel" name="label_3">
130                 <property name="text">
131                     <string>音频格式</string>
132                 </property>

```

```
132     </widget>
133 </item>
134 <item>
135     <widget class="QRadioButton" name="radioButton_mp3">
136         <property name="enabled">
137             <bool>false</bool>
138         </property>
139         <property name="text">
140             <string>mp3</string>
141         </property>
142     </widget>
143 </item>
144 <item>
145     <widget class="QRadioButton" name="radioButton_wav">
146         <property name="enabled">
147             <bool>false</bool>
148         </property>
149         <property name="text">
150             <string>wav</string>
151         </property>
152     </widget>
153 </item>
154 </layout>
155 </widget>
156 <widget class="QWidget" name="layoutWidget">
157     <property name="geometry">
158         <rect>
159             <x>40</x>
160             <y>130</y>
161             <width>291</width>
162             <height>22</height>
163         </rect>
164     </property>
165     <layout class="QHBoxLayout" name="horizontalLayout_4">
166         <item>
167             <widget class="QLabel" name="label_2">
168                 <property name="text">
169                     <string>视频格式</string>
170                 </property>
```

```
171     </widget>
172 </item>
173 <item>
174     <widget class="QRadioButton" name="radioButton_mp4">
175         <property name="text">
176             <string>mp4</string>
177         </property>
178     </widget>
179 </item>
180 <item>
181     <widget class="QRadioButton" name="radioButton_avi">
182         <property name="text">
183             <string>avi</string>
184         </property>
185     </widget>
186 </item>
187 </layout>
188 </widget>
189 </widget>
190 <resources/>
191 <connections/>
192 </ui>
```

1.8 child_setting.py

```
1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file
4  ↳ 'child_setting.ui'
5  #
6  # Created by: PyQt5 UI code generator 5.15.10
7  #
8  # WARNING: Any manual changes made to this file will be lost
9  ↳ when pyuic5 is
10 # run again. Do not edit this file unless you know what you
11 ↳ are doing.
```

```

185     <property name="sizeHint" stdset="0">
186     <size>
187         <width>40</width>
188         <height>20</height>
189     </size>
190 </property>
191 </spacer>
192 </item>
193 <item>
194     <layout class="QVBoxLayout" name="verticalLayout_5">
195     <item>
196         <layout class="QVBoxLayout" name="verticalLayout_6">
197         <item>
198             <layout class="QHBoxLayout"
199             ↪ name="horizontalLayout_8">
200             <item>
201                 <spacer name="horizontalSpacer_16">
202                 <property name="orientation">
203                     <enum>Qt::Horizontal</enum>
204                 </property>
205                 <property name="sizeType">
206                     <enum>QSizePolicy::Maximum</enum>
207                 </property>
208                 <property name="sizeHint" stdset="0">
209                 <size>
210                     <width>30</width>
211                     <height>20</height>
212                 </size>
213                 </property>
214             </spacer>
215         </item>
216         <item>
217             <widget class="QComboBox" name="tab1_combobox_3">
218                 <property name="sizePolicy">
219                     <sizepolicy hsizeType="MinimumExpanding"
220                     ↪ vsizeType="Fixed">
221                     <horstretch>0</horstretch>
222                     <verstretch>0</verstretch>
223                 </sizepolicy>

```

```
222         </property>
223     </widget>
224 </item>
225 <item>
226     <spacer name="horizontalSpacer_17">
227         <property name="orientation">
228             <enum>Qt::Horizontal</enum>
229         </property>
230         <property name="sizeType">
231             <enum>QSizePolicy::Maximum</enum>
232         </property>
233         <property name="sizeHint" stdset="0">
234             <size>
235                 <width>30</width>
236                 <height>20</height>
237             </size>
238         </property>
239     </spacer>
240 </item>
241 </layout>
242 </item>
243 <item>
244     <spacer name="horizontalSpacer_18">
245         <property name="orientation">
246             <enum>Qt::Horizontal</enum>
247         </property>
248         <property name="sizeHint" stdset="0">
249             <size>
250                 <width>40</width>
251                 <height>20</height>
252             </size>
253         </property>
254     </spacer>
255 </item>
256 </layout>
257 </item>
258 <item>
259     <widget class="QLabel" name="tab1_label_3">
260         <property name="sizePolicy">
```



```
261         <sizepolicy hsize="Ignored" vsize="Fixed">
262             <horstretch>0</horstretch>
263             <verstretch>0</verstretch>
264         </sizepolicy>
265     </property>
266     <property name="minimumSize">
267         <size>
268             <width>256</width>
269             <height>144</height>
270         </size>
271     </property>
272     <property name="maximumSize">
273         <size>
274             <width>256</width>
275             <height>144</height>
276         </size>
277     </property>
278     <property name="text">
279         <string>TextLabel</string>
280     </property>
281     <property name="alignment">
282         <set>Qt::AlignCenter</set>
283     </property>
284 </widget>
285 </item>
286 </layout>
287 </item>
288 <item>
289     <spacer name="horizontalSpacer_19">
290         <property name="orientation">
291             <enum>Qt::Horizontal</enum>
292         </property>
293         <property name="sizeHint" stdset="0">
294             <size>
295                 <width>40</width>
296                 <height>20</height>
297             </size>
298         </property>
299     </spacer>
```

```

300         </item>
301     </layout>
302 </widget>
303 </widget>
304 <widget class="QWidget" name="tab_3">
305     <attribute name="title">
306         <string>眼睛</string>
307     </attribute>
308     <widget class="QWidget" name="layoutWidget_3">
309         <property name="geometry">
310             <rect>
311                 <x>0</x>
312                 <y>10</y>
313                 <width>371</width>
314                 <height>211</height>
315             </rect>
316         </property>
317         <layout class="QHBoxLayout" name="horizontalLayout_4">
318             <item>
319                 <spacer name="horizontalSpacer_20">
320                     <property name="orientation">
321                         <enum>Qt::Horizontal</enum>
322                     </property>
323                     <property name="sizeHint" stdset="0">
324                         <size>
325                             <width>40</width>
326                             <height>20</height>
327                         </size>
328                     </property>
329                 </spacer>
330             </item>
331             <item>
332                 <layout class="QVBoxLayout" name="verticalLayout_7">
333                     <item>
334                         <layout class="QVBoxLayout" name="verticalLayout_8">
335                             <item>
336                                 <layout class="QHBoxLayout"
337                                     ↪ name="horizontalLayout_9">
338                                     <item>

```

```
338     <spacer name="horizontalSpacer_5">
339         <property name="orientation">
340             <enum>Qt::Horizontal</enum>
341         </property>
342         <property name="sizeType">
343             <enum>QSizePolicy::Maximum</enum>
344         </property>
345         <property name="sizeHint" stdset="0">
346             <size>
347                 <width>30</width>
348                 <height>20</height>
349             </size>
350         </property>
351     </spacer>
352 </item>
353 <item>
354     <widget class="QComboBox" name="tab1_combobox_4">
355         <property name="sizePolicy">
356             <sizepolicy hsizeType="MinimumExpanding"
357                 ↪ vsizeType="Fixed">
358                 <horstretch>0</horstretch>
359                 <verstretch>0</verstretch>
360             </sizepolicy>
361         </property>
362     </widget>
363 </item>
364 <item>
365     <spacer name="horizontalSpacer_6">
366         <property name="orientation">
367             <enum>Qt::Horizontal</enum>
368         </property>
369         <property name="sizeType">
370             <enum>QSizePolicy::Maximum</enum>
371         </property>
372         <property name="sizeHint" stdset="0">
373             <size>
374                 <width>30</width>
375                 <height>20</height>
376             </size>
```

```
376         </property>
377     </spacer>
378 </item>
379 </layout>
380 </item>
381 <item>
382     <spacer name="horizontalSpacer_21">
383         <property name="orientation">
384             <enum>Qt::Horizontal</enum>
385         </property>
386         <property name="sizeHint" stdset="0">
387             <size>
388                 <width>40</width>
389                 <height>20</height>
390             </size>
391         </property>
392     </spacer>
393 </item>
394 </layout>
395 </item>
396 <item>
397     <widget class="QLabel" name="tab1_label_4">
398         <property name="sizePolicy">
399             <sizepolicy hstretch="Ignored" vstretch="Fixed">
400                 <horstretch>0</horstretch>
401                 <verstretch>0</verstretch>
402             </sizepolicy>
403         </property>
404         <property name="minimumSize">
405             <size>
406                 <width>256</width>
407                 <height>144</height>
408             </size>
409         </property>
410         <property name="maximumSize">
411             <size>
412                 <width>256</width>
413                 <height>144</height>
414             </size>
```

```
415         </property>
416         <property name="text">
417             <string>TextLabel</string>
418         </property>
419         <property name="alignment">
420             <set>Qt::AlignCenter</set>
421         </property>
422     </widget>
423 </item>
424 </layout>
425 </item>
426 <item>
427     <spacer name="horizontalSpacer_22">
428         <property name="orientation">
429             <enum>Qt::Horizontal</enum>
430         </property>
431         <property name="sizeHint" stdset="0">
432             <size>
433                 <width>40</width>
434                 <height>20</height>
435             </size>
436         </property>
437     </spacer>
438 </item>
439 </layout>
440 </widget>
441 </widget>
442 <widget class="QWidget" name="tab_4">
443     <attribute name="title">
444         <string>麦克风</string>
445     </attribute>
446     <widget class="QWidget" name="layoutWidget">
447         <property name="geometry">
448             <rect>
449                 <x>0</x>
450                 <y>10</y>
451                 <width>371</width>
452                 <height>211</height>
453             </rect>
```

```
454     </property>
455     <layout class="QHBoxLayout" name="horizontalLayout_13">
456         <item>
457             <spacer name="horizontalSpacer_31">
458                 <property name="orientation">
459                     <enum>Qt::Horizontal</enum>
460                 </property>
461                 <property name="sizeHint" stdset="0">
462                     <size>
463                         <width>40</width>
464                         <height>40</height>
465                     </size>
466                 </property>
467             </spacer>
468         </item>
469         <item>
470             <layout class="QVBoxLayout" name="verticalLayout_13">
471                 <item>
472                     <layout class="QVBoxLayout" name="verticalLayout_14">
473                         <item>
474                             <layout class="QHBoxLayout"
475                             ↪ name="horizontalLayout_14">
476                                 <item>
477                                     <spacer name="horizontalSpacer_32">
478                                         <property name="orientation">
479                                             <enum>Qt::Horizontal</enum>
480                                         </property>
481                                         <property name="sizeType">
482                                             <enum>QSizePolicy::Maximum</enum>
483                                         </property>
484                                         <property name="sizeHint" stdset="0">
485                                             <size>
486                                                 <width>31</width>
487                                                 <height>20</height>
488                                             </size>
489                                         </property>
490                                     </spacer>
491                                 </item>
492                                 <item>
```

```
492     <widget class="QComboBox" name="tab1_combobox_7">
493         <property name="sizePolicy">
494             <sizepolicy hsize="MinimumExpanding"
495                 ↪ vsize="Fixed">
496                 <horstretch>0</horstretch>
497                 <verstretch>0</verstretch>
498             </sizepolicy>
499         </property>
500         <property name="minimumSize">
501             <size>
502                 <width>160</width>
503                 <height>0</height>
504             </size>
505         </property>
506     </widget>
507 </item>
508 <item>
509     <spacer name="horizontalSpacer_33">
510         <property name="orientation">
511             <enum>Qt::Horizontal</enum>
512         </property>
513         <property name="sizeType">
514             <enum>QSizePolicy::Maximum</enum>
515         </property>
516         <property name="sizeHint" stdset="0">
517             <size>
518                 <width>31</width>
519                 <height>20</height>
520             </size>
521         </property>
522     </spacer>
523 </item>
524 </layout>
525 </item>
526 <item>
527     <spacer name="horizontalSpacer_34">
528         <property name="orientation">
529             <enum>Qt::Horizontal</enum>
530         </property>
```

```
530         <property name="sizeType">
531             <enum>QSizePolicy::Minimum</enum>
532         </property>
533         <property name="sizeHint" stdset="0">
534             <size>
535                 <width>256</width>
536                 <height>55</height>
537             </size>
538         </property>
539     </spacer>
540 </item>
541 </layout>
542 </item>
543 <item>
544     <widget class="QProgressBar" name="progressBar">
545         <property name="value">
546             <number>0</number>
547         </property>
548         <property name="textVisible">
549             <bool>false</bool>
550         </property>
551     </widget>
552 </item>
553 <item>
554     <spacer name="horizontalSpacer_36">
555         <property name="orientation">
556             <enum>Qt::Horizontal</enum>
557         </property>
558         <property name="sizeHint" stdset="0">
559             <size>
560                 <width>40</width>
561                 <height>60</height>
562             </size>
563         </property>
564     </spacer>
565 </item>
566 </layout>
567 </item>
568 <item>
```



```
569     <spacer name="horizontalSpacer_35">
570     <property name="orientation">
571     <enum>Qt::Horizontal</enum>
572     </property>
573     <property name="sizeHint" stdset="0">
574     <size>
575     <width>40</width>
576     <height>20</height>
577     </size>
578     </property>
579     </spacer>
580 </item>
581 </layout>
582 </widget>
583 </widget>
584 </widget>
585 <widget class="QPushButton" name="OK">
586 <property name="geometry">
587 <rect>
588 <x>260</x>
589 <y>280</y>
590 <width>111</width>
591 <height>23</height>
592 </rect>
593 </property>
594 <property name="text">
595 <string>确认</string>
596 </property>
597 </widget>
598 </widget>
599 <resources/>
600 <connections/>
601 </ui>
```

1.12 gui.py

```
1 # -*- coding: utf-8 -*-
2
```

```
3  # Form implementation generated from reading ui file 'gui.ui'
4  #
5  # Created by: PyQt5 UI code generator 5.15.10
6  #
7  # WARNING: Any manual changes made to this file will be lost
8  # when pyuic5 is
9  # run again. Do not edit this file unless you know what you
10 # are doing.
11
12
13
14 from PyQt5 import QtCore, QtGui, QtWidgets
15
16
17
18 class Ui_MainWindow(object):
19     def setupUi(self, MainWindow):
20         MainWindow.setObjectName("MainWindow")
21         MainWindow.resize(1048, 1025)
22         icon = QtGui.QIcon()
23         icon.addPixmap(QtGui.QPixmap("../icon.ico"),
24             QtGui.QIcon.Normal, QtGui.QIcon.Off)
25         MainWindow.setWindowIcon(icon)
26         self.centralwidget = QtWidgets.QWidget(MainWindow)
27         self.centralwidget.setObjectName("centralwidget")
28         self.verticalLayout_4 =
29             QtWidgets.QVBoxLayout(self.centralwidget)
30         self.verticalLayout_4.setObjectName("verticalLayout_4",
31             ")
32         self.h_img_lay = QtWidgets.QHBoxLayout()
33         self.h_img_lay.setObjectName("h_img_lay")
34         self.original_img_v_lay = QtWidgets.QVBoxLayout()
35         self.original_img_v_lay.setObjectName("original_img_v_lay",
36             "_lay")
37         spacerItem = QtWidgets.QSpacerItem(20, 40,
38             QtWidgets.QSizePolicy.Minimum,
39             QtWidgets.QSizePolicy.Expanding)
40         self.original_img_v_lay.addItem(spacerItem)
41         self.original_name_lbl =
42             QtWidgets.QLabel(self.centralwidget)
43         self.original_name_lbl.setScaledContents(False)
```

```
33         self.original_name_lbl.setAlignment(QtCore.Qt.AlignCe  
        ↪ nter)  
34         self.original_name_lbl.setObjectName("original_name_l  
        ↪ bl")  
35         self.original_img_v_lay.addWidget(self.original_name_  
        ↪ lbl)  
36         spacerItem1 = QtWidgets.QSpacerItem(20, 40,  
        ↪ QtWidgets.QSizePolicy.Minimum,  
        ↪ QtWidgets.QSizePolicy.Expanding)  
37         self.original_img_v_lay.addItem(spacerItem1)  
38         self.original_frame_lbl =  
        ↪ QtWidgets.QLabel(self.centralwidget)  
39         self.original_frame_lbl.setMinimumSize(QtCore.QSize(5  
        ↪ 00,  
        ↪ 300))  
40         self.original_frame_lbl.setObjectName("original_frame  
        ↪ _lbl")  
41         self.original_img_v_lay.addWidget(self.original_frame_  
        ↪ _lbl)  
42         self.selecamera1 =  
        ↪ QtWidgets.QComboBox(self.centralwidget)  
43         self.selecamera1.setObjectName("selecamera1")  
44         self.original_img_v_lay.addWidget(self.selecamera1)  
45         spacerItem2 = QtWidgets.QSpacerItem(20, 40,  
        ↪ QtWidgets.QSizePolicy.Minimum,  
        ↪ QtWidgets.QSizePolicy.Expanding)  
46         self.original_img_v_lay.addItem(spacerItem2)  
47         self.h_img_lay.addLayout(self.original_img_v_lay)  
48         self.processed_img_v_lay = QtWidgets.QVBoxLayout()  
49         self.processed_img_v_lay.setObjectName("processed_img_  
        ↪ _v_lay")  
50         spacerItem3 = QtWidgets.QSpacerItem(20, 40,  
        ↪ QtWidgets.QSizePolicy.Minimum,  
        ↪ QtWidgets.QSizePolicy.Expanding)  
51         self.processed_img_v_lay.addItem(spacerItem3)  
52         self.processed_name_lbl =  
        ↪ QtWidgets.QLabel(self.centralwidget)  
53         self.processed_name_lbl.setAlignment(QtCore.Qt.AlignC  
        ↪ enter)
```

```
54     self.processed_name_lbl.setObjectName("processed_name_1  
    ↪ _lbl")  
55     self.processed_img_v_layout.addWidget(self.processed_name_1  
    ↪ e_lbl)  
56     spacerItem4 = QtWidgets.QSpacerItem(20, 40,  
    ↪ QtWidgets.QSizePolicy.Minimum,  
    ↪ QtWidgets.QSizePolicy.Expanding)  
57     self.processed_img_v_layout.addItem(spacerItem4)  
58     self.processed_frame_lbl =  
    ↪ QtWidgets.QLabel(self.centralwidget)  
59     self.processed_frame_lbl.setMinimumSize(QtCore.QSize(1  
    ↪ 500,  
    ↪ 300))  
60     self.processed_frame_lbl.setBaseSize(QtCore.QSize(0,  
    ↪ 0))  
61     self.processed_frame_lbl.setText("")  
62     self.processed_frame_lbl.setObjectName("processed_frame_1  
    ↪ me_lbl")  
63     self.processed_img_v_layout.addWidget(self.processed_frame_1  
    ↪ me_lbl)  
64     self.selecamera2 =  
    ↪ QtWidgets.QComboBox(self.centralwidget)  
65     self.selecamera2.setObjectName("selecamera2")  
66     self.processed_img_v_layout.addWidget(self.selecamera2)  
67     spacerItem5 = QtWidgets.QSpacerItem(20, 40,  
    ↪ QtWidgets.QSizePolicy.Minimum,  
    ↪ QtWidgets.QSizePolicy.Expanding)  
68     self.processed_img_v_layout.addItem(spacerItem5)  
69     self.h_img_layout.addLayout(self.processed_img_v_layout)  
70     self.verticalLayout_4.addLayout(self.h_img_layout)  
71     self.horizontalLayout = QtWidgets.QHBoxLayout()  
72     self.horizontalLayout.setObjectName("horizontalLayout_1  
    ↪ ")  
73     self.verticalLayout_2 = QtWidgets.QVBoxLayout()  
74     self.verticalLayout_2.setObjectName("verticalLayout_2_1  
    ↪ ")  
75     self.label_3 = QtWidgets.QLabel(self.centralwidget)  
76     self.label_3.setObjectName("label_3")  
77     self.verticalLayout_2.addWidget(self.label_3)
```

```
78         self.eye_lbl = QtWidgets.QLabel(self.centralwidget)
79         self.eye_lbl.setMinimumSize(QtCore.QSize(500, 300))
80         self.eye_lbl.setText("")
81         self.eye_lbl.setObjectName("eye_lbl")
82         self.verticalLayout_2.addWidget(self.eye_lbl)
83         self.comboBox_2 =
84             ↳ QtWidgets.QComboBox(self.centralwidget)
85         self.comboBox_2.setObjectName("comboBox_2")
86         self.verticalLayout_2.addWidget(self.comboBox_2)
87         self.horizontalLayout.addLayout(self.verticalLayout_2)
88         self.verticalLayout = QtWidgets.QVBoxLayout()
89         self.verticalLayout.setObjectName("verticalLayout")
90         self.label_2 = QtWidgets.QLabel(self.centralwidget)
91         self.label_2.setObjectName("label_2")
92         self.verticalLayout.addWidget(self.label_2)
93         self.voice_lbl = QtWidgets.QLabel(self.centralwidget)
94         self.voice_lbl.setMinimumSize(QtCore.QSize(500, 300))
95         self.voice_lbl.setText("")
96         self.voice_lbl.setObjectName("voice_lbl")
97         self.verticalLayout.addWidget(self.voice_lbl)
98         self.comboBox =
99             ↳ QtWidgets.QComboBox(self.centralwidget)
100         self.comboBox.setObjectName("comboBox")
101         self.verticalLayout.addWidget(self.comboBox)
102         self.horizontalLayout.addLayout(self.verticalLayout)
103         self.verticalLayout_4.addLayout(self.horizontalLayout)
104         self.h_btn_lay = QtWidgets.QHBoxLayout()
105         self.h_btn_lay.setObjectName("h_btn_lay")
106         self.initButton =
107             ↳ QtWidgets.QPushButton(self.centralwidget)
108         self.initButton.setObjectName("initButton")
109         self.h_btn_lay.addWidget(self.initButton)
110         self.pushButton =
111             ↳ QtWidgets.QPushButton(self.centralwidget)
112         self.pushButton.setObjectName("pushButton")
113         self.h_btn_lay.addWidget(self.pushButton)
114         spacerItem6 = QtWidgets.QSpacerItem(40, 20,
115             ↳ QtWidgets.QSizePolicy.Minimum,
116             ↳ QtWidgets.QSizePolicy.Expanding)
```

```
111         self.h_btn_layout.addItem(spacerItem6)
112         self.label = QtWidgets.QLabel(self.centralwidget)
113         self.label.setObjectName("label")
114         self.h_btn_layout.addWidget(self.label)
115         self.lineEdit =
116             ↳ QtWidgets.QLineEdit(self.centralwidget)
117         self.lineEdit.setObjectName("lineEdit")
118         self.h_btn_layout.addWidget(self.lineEdit)
119         self.startButton =
120             ↳ QtWidgets.QPushButton(self.centralwidget)
121         self.startButton.setObjectName("startButton")
122         self.h_btn_layout.addWidget(self.startButton)
123         self.endButton =
124             ↳ QtWidgets.QPushButton(self.centralwidget)
125         self.endButton.setObjectName("endButton")
126         self.h_btn_layout.addWidget(self.endButton)
127         self.shotButton =
128             ↳ QtWidgets.QPushButton(self.centralwidget)
129         self.shotButton.setObjectName("shotButton")
130         self.h_btn_layout.addWidget(self.shotButton)
131         self.verticalLayout_4.addLayout(self.h_btn_layout)
132         self.v_filters_layout = QtWidgets.QVBoxLayout()
133         self.v_filters_layout.setObjectName("v_filters_layout")
134         self.verticalLayout_4.addLayout(self.v_filters_layout)
135         MainWindow.setCentralWidget(self.centralwidget)
136         self.menubar = QtWidgets.QMenuBar(MainWindow)
137         self.menubar.setGeometry(QtCore.QRect(0, 0, 1048, 37))
138         self.menubar.setObjectName("menubar")
139         self.menuHelp = QtWidgets.QMenu(self.menubar)
140         self.menuHelp.setObjectName("menuHelp")
141         self.menu = QtWidgets.QMenu(self.menubar)
142         self.menu.setObjectName("menu")
143         self.menu_2 = QtWidgets.QMenu(self.menubar)
144         self.menu_2.setObjectName("menu_2")
145         MainWindow.setMenuBar(self.menubar)
146         self.statusbar = QtWidgets.QStatusBar(MainWindow)
147         self.statusbar.setMinimumSize(QtCore.QSize(500, 30))
148         self.statusbar.setObjectName("statusbar")
149         MainWindow.setStatusBar(self.statusbar)
```

```
146         self.actionOpen_image = QtWidgets.QAction(MainWindow)
147         self.actionOpen_image.setObjectName("actionOpen_image")
148         self.actionSave_original_image =
149             QtWidgets.QAction(MainWindow)
150         self.actionSave_original_image.setObjectName("actionSave_
151             original_image")
152         self.actionSave_processed_image =
153             QtWidgets.QAction(MainWindow)
154         self.actionSave_processed_image.setObjectName("action_
155             Save_processed_image")
156         self.actionExit = QtWidgets.QAction(MainWindow)
157         self.actionExit.setObjectName("actionExit")
158         self.actionLicense = QtWidgets.QAction(MainWindow)
159         self.actionLicense.setObjectName("actionLicense")
160         self.actionAbout = QtWidgets.QAction(MainWindow)
161         self.actionAbout.setObjectName("actionAbout")
162         self.actionmanual = QtWidgets.QAction(MainWindow)
163         self.actionmanual.setObjectName("actionmanual")
164         self.action12 = QtWidgets.QAction(MainWindow)
165         self.action12.setObjectName("action12")
166         self.actionse = QtWidgets.QAction(MainWindow)
167         self.actionse.setObjectName("actionse")
168         self.menuHelp.addAction(self.actionmanual)
169         self.menuHelp.addAction(self.actionAbout)
170         self.menu_2.addAction(self.action12)
171         self.menu_2.addAction(self.actionse)
172         self.menubar.addAction(self.menu.menuAction())
173         self.menubar.addAction(self.menu_2.menuAction())
174         self.menubar.addAction(self.menuHelp.menuAction())
175
176         self.retranslateUi(MainWindow)
177         QtCore.QMetaObject.connectSlotsByName(MainWindow)
178
179     def retranslateUi(self, MainWindow):
180         _translate = QtCore.QCoreApplication.translate
181         MainWindow.setWindowTitle(_translate("MainWindow",
182             "面部表型特征提取系统---山东省精神卫生中心"))
```

```

178         self.original_name_lbl.setText(_translate("MainWindow",
            ↳ "",
            ↳ "原始图像"))
179         self.original_frame_lbl.setText(_translate("MainWindo",
            ↳ "w",
            ↳ "<html><head/><body><p><br/></p></body></html>"))
180         self.processed_name_lbl.setText(_translate("MainWindo",
            ↳ "w",
            ↳ "面部特征"))
181         self.label_3.setText(_translate("MainWindow",
            ↳ "标记图像"))
182         self.label_2.setText(_translate("MainWindow",
            ↳ "情绪识别"))
183         self.initButton.setText(_translate("MainWindow",
            ↳ "检测视频设备"))
184         self.pushButton.setText(_translate("MainWindow",
            ↳ "停用视频设备"))
185         self.label.setText(_translate("MainWindow", "样本编号:
            ↳ "))
186         self.startButton.setText(_translate("MainWindow",
            ↳ "开始提取"))
187         self.endButton.setText(_translate("MainWindow",
            ↳ "结束提取"))
188         self.shotButton.setText(_translate("MainWindow",
            ↳ "视频截图"))
189         self.menuHelp.setTitle(_translate("MainWindow",
            ↳ "帮助"))
190         self.menu.setTitle(_translate("MainWindow", " 文件"))
191         self.menu_2.setTitle(_translate("MainWindow", "设置"))
192         self.actionOpen_image.setText(_translate("MainWindow",
            ↳ "Open image"))
193         self.actionSave_original_image.setText(_translate("Ma",
            ↳ "inWindow", "Save original
            ↳ image"))
194         self.actionSave_processed_image.setText(_translate("M",
            ↳ "ainWindow", "Save processed
            ↳ image"))
195         self.actionExit.setText(_translate("MainWindow",
            ↳ "Exit"))

```



```

196         self.actionLicense.setText(_translate("MainWindow",
        ↪ "License"))
197         self.actionAbout.setText(_translate("MainWindow",
        ↪ "关于"))
198         self.actionmanual.setText(_translate("MainWindow",
        ↪ "操作说明"))
199         self.action12.setText(_translate("MainWindow",
        ↪ "设置视频与音频设备"))
200         self.actionse.setText(_translate("MainWindow",
        ↪ "设置文件保存路径"))
201
202
203 if __name__ == "__main__":
204     import sys
205     app = QtWidgets.QApplication(sys.argv)
206     MainWindow = QtWidgets.QMainWindow()
207     ui = Ui_MainWindow()
208     ui.setupUi(MainWindow)
209     MainWindow.show()
210     sys.exit(app.exec_())

```

1.13 gui.ui

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3     <class>MainWindow</class>
4     <widget class="QMainWindow" name="MainWindow">
5         <property name="geometry">
6             <rect>
7                 <x>0</x>
8                 <y>0</y>
9                 <width>1048</width>
10                <height>1025</height>
11            </rect>
12        </property>
13        <property name="windowTitle">
14            <string>面部表型特征提取系统---山东省精神卫生中心</string>
15        </property>

```

```
16 <property name="windowIcon">
17   <iconset>
18     <normaloff>../icon.ico</normaloff>../icon.ico</iconset>
19 </property>
20 <widget class="QWidget" name="centralwidget">
21   <layout class="QVBoxLayout" name="verticalLayout_4">
22     <item>
23       <layout class="QHBoxLayout" name="h_img_lay">
24         <item>
25           <layout class="QVBoxLayout" name="original_img_v_lay">
26             <item>
27               <spacer name="verticalSpacer">
28                 <property name="sizeHint" stdset="0">
29                   <size>
30                     <width>20</width>
31                     <height>40</height>
32                   </size>
33                 </property>
34               </spacer>
35             </item>
36             <item>
37               <widget class="QLabel" name="original_name_lbl">
38                 <property name="text">
39                   <string>原始图像</string>
40                 </property>
41                 <property name="scaledContents">
42                   <bool>false</bool>
43                 </property>
44                 <property name="alignment">
45                   <set>Qt::AlignCenter</set>
46                 </property>
47               </widget>
48             </item>
49             <item>
50               <spacer name="verticalSpacer_2">
51                 <property name="sizeHint" stdset="0">
52                   <size>
53                     <width>20</width>
54                     <height>40</height>
```

```

55         </size>
56     </property>
57 </spacer>
58 </item>
59 <item>
60     <widget class="QLabel" name="original_frame_lbl">
61         <property name="minimumSize">
62             <size>
63                 <width>500</width>
64                 <height>300</height>
65             </size>
66         </property>
67         <property name="text">
68             <string>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p
        ↪ &gt;&lt;br/&gt;&lt;/p&gt;&lt;/body&gt;&lt;/html
        ↪ l&gt;</string>
69         </property>
70     </widget>
71 </item>
72 <item>
73     <widget class="QComboBox" name="selecamera1"/>
74 </item>
75 <item>
76     <spacer name="verticalSpacer_3">
77         <property name="sizeHint" stdset="0">
78             <size>
79                 <width>20</width>
80                 <height>40</height>
81             </size>
82         </property>
83     </spacer>
84 </item>
85 </layout>
86 </item>
87 <item>
88     <layout class="QVBoxLayout" name="processed_img_v_lay">
89         <item>
90             <spacer name="verticalSpacer_4">
91                 <property name="sizeHint" stdset="0">

```

```
92         <size>
93             <width>20</width>
94             <height>40</height>
95         </size>
96     </property>
97 </spacer>
98 </item>
99 <item>
100     <widget class="QLabel" name="processed_name_lbl">
101         <property name="text">
102             <string>面部特征</string>
103         </property>
104         <property name="alignment">
105             <set>Qt::AlignCenter</set>
106         </property>
107     </widget>
108 </item>
109 <item>
110     <spacer name="verticalSpacer_5">
111         <property name="sizeHint" stdset="0">
112             <size>
113                 <width>20</width>
114                 <height>40</height>
115             </size>
116         </property>
117     </spacer>
118 </item>
119 <item>
120     <widget class="QLabel" name="processed_frame_lbl">
121         <property name="minimumSize">
122             <size>
123                 <width>500</width>
124                 <height>300</height>
125             </size>
126         </property>
127         <property name="baseSize">
128             <size>
129                 <width>0</width>
130                 <height>0</height>
```

```
131         </size>
132     </property>
133     <property name="text">
134         <string/>
135     </property>
136 </widget>
137 </item>
138 <item>
139     <widget class="QComboBox" name="selecamera2"/>
140 </item>
141 <item>
142     <spacer name="verticalSpacer_6">
143         <property name="sizeHint" stdset="0">
144             <size>
145                 <width>20</width>
146                 <height>40</height>
147             </size>
148         </property>
149     </spacer>
150 </item>
151 </layout>
152 </item>
153 </layout>
154 </item>
155 <item>
156     <layout class="QHBoxLayout" name="horizontalLayout">
157         <item>
158             <layout class="QVBoxLayout" name="verticalLayout_2">
159                 <item>
160                     <widget class="QLabel" name="label_3">
161                         <property name="text">
162                             <string>标记图像</string>
163                         </property>
164                     </widget>
165                 </item>
166                 <item>
167                     <widget class="QLabel" name="eye_lbl">
168                         <property name="minimumSize">
169                             <size>
```

```
170         <width>500</width>
171         <height>300</height>
172     </size>
173     </property>
174     <property name="text">
175         <string/>
176     </property>
177 </widget>
178 </item>
179 <item>
180     <widget class="QComboBox" name="comboBox_2"/>
181 </item>
182 </layout>
183 </item>
184 <item>
185     <layout class="QVBoxLayout" name="verticalLayout">
186     <item>
187         <widget class="QLabel" name="label_2">
188             <property name="text">
189                 <string>情绪识别</string>
190             </property>
191         </widget>
192     </item>
193     <item>
194         <widget class="QLabel" name="voice_lbl">
195             <property name="minimumSize">
196                 <size>
197                     <width>500</width>
198                     <height>300</height>
199                 </size>
200             </property>
201             <property name="text">
202                 <string/>
203             </property>
204         </widget>
205     </item>
206 </item>
207     <widget class="QComboBox" name="comboBox"/>
208 </item>
```

```
209         </layout>
210     </item>
211 </layout>
212 </item>
213 <item>
214     <layout class="QHBoxLayout" name="h_btn_lay">
215         <item>
216             <widget class="QPushButton" name="initButton">
217                 <property name="text">
218                     <string>检测视频设备</string>
219                 </property>
220             </widget>
221         </item>
222         <item>
223             <widget class="QPushButton" name="pushButton">
224                 <property name="text">
225                     <string>停用视频设备</string>
226                 </property>
227             </widget>
228         </item>
229         <item>
230             <spacer name="horizontalSpacer_2">
231                 <property name="sizeHint" stdset="0">
232                     <size>
233                         <width>40</width>
234                         <height>20</height>
235                     </size>
236                 </property>
237             </spacer>
238         </item>
239         <item>
240             <widget class="QLabel" name="label">
241                 <property name="text">
242                     <string>样本编号: </string>
243                 </property>
244             </widget>
245         </item>
246         <item>
247             <widget class="QLineEdit" name="lineEdit"/>
```

```
248     </item>
249     <item>
250         <widget class="QPushButton" name="startButton">
251             <property name="text">
252                 <string>开始提取</string>
253             </property>
254         </widget>
255     </item>
256     <item>
257         <widget class="QPushButton" name="endButton">
258             <property name="text">
259                 <string>结束提取</string>
260             </property>
261         </widget>
262     </item>
263     <item>
264         <widget class="QPushButton" name="shotButton">
265             <property name="text">
266                 <string>视频截图</string>
267             </property>
268         </widget>
269     </item>
270 </layout>
271 </item>
272 <item>
273     <layout class="QVBoxLayout" name="v_filters_lay"/>
274 </item>
275 </layout>
276 </widget>
277 <widget class="QMenuBar" name="menubar">
278     <property name="geometry">
279         <rect>
280             <x>0</x>
281             <y>0</y>
282             <width>1048</width>
283             <height>37</height>
284         </rect>
285     </property>
286     <widget class="QMenu" name="menuHelp">
```



```
287     <property name="title">
288         <string>帮助</string>
289     </property>
290     <addaction name="actionmanual"/>
291     <addaction name="actionAbout"/>
292 </widget>
293 <widget class="QMenu" name="menu">
294     <property name="title">
295         <string>文件</string>
296     </property>
297 </widget>
298 <widget class="QMenu" name="menu_2">
299     <property name="title">
300         <string>设置</string>
301     </property>
302     <addaction name="action12"/>
303     <addaction name="actionse"/>
304 </widget>
305     <addaction name="menu"/>
306     <addaction name="menu_2"/>
307     <addaction name="menuHelp"/>
308 </widget>
309 <widget class="QStatusBar" name="statusbar">
310     <property name="minimumSize">
311         <size>
312             <width>500</width>
313             <height>30</height>
314         </size>
315     </property>
316 </widget>
317 <action name="actionOpen_image">
318     <property name="text">
319         <string>Open image</string>
320     </property>
321 </action>
322 <action name="actionSave_original_image">
323     <property name="text">
324         <string>Save original image</string>
325     </property>
```

```
326     </action>
327     <action name="actionSave_processed_image">
328         <property name="text">
329             <string>Save processed image</string>
330         </property>
331     </action>
332     <action name="actionExit">
333         <property name="text">
334             <string>Exit</string>
335         </property>
336     </action>
337     <action name="actionLicense">
338         <property name="text">
339             <string>License</string>
340         </property>
341     </action>
342     <action name="actionAbout">
343         <property name="text">
344             <string>关于</string>
345         </property>
346     </action>
347     <action name="actionmanual">
348         <property name="text">
349             <string>操作说明</string>
350         </property>
351     </action>
352     <action name="action12">
353         <property name="text">
354             <string>设置视频与音频设备</string>
355         </property>
356     </action>
357     <action name="actionse">
358         <property name="text">
359             <string>设置文件保存路径</string>
360         </property>
361     </action>
362 </widget>
363 <resources/>
364 <connections/>
```

365 </ui>

1.14 main.py

```

1  if __name__ == '__main__':
2      import sys,platform
3      from PyQt5.QtWidgets import QApplication
4      #from CvPyGui import Main
5      from CvPyGui import main as Main
6      if platform.system() == "Windows":
7          import ctypes
8          ctypes.windll.shell32.SetCurrentProcessExplicitAppUserModelID("myappid")
9      app = QApplication(sys.argv)
10     window = Main.MyApp()
11     window.show()
12     global update1
13     update1 = 0
14     global update2
15     update2 = 0
16     sys.exit(app.exec_())

```

1.15 test.py

```

1  #!/usr/bin/env python
2  #! -*- coding:utf-8 -*-
3  import cv2
4  import time
5
6  if __name__ == '__main__' :
7      # 启动默认相机
8      video = cv2.VideoCapture(0);
9
10     # 获取 OpenCV version
11     (major_ver, minor_ver, subminor_ver) =
12         (cv2.__version__).split('.')

```

```
13     # 对于 webcam 不能采用 get(CV_CAP_PROP_FPS) 方法
14     # 而是:
15     if int(major_ver) < 3 :
16         fps = video.get(cv2.cv.CV_CAP_PROP_FPS)
17         print("Frames per second using
            ↳ video.get(cv2.cv.CV_CAP_PROP_FPS):
            ↳ {0}".format(fps))
18     else :
19         fps = video.get(cv2.CAP_PROP_FPS)
20         print("Frames per second using
            ↳ video.get(cv2.CAP_PROP_FPS) : {0}".format(fps))
21
22     # Number of frames to capture
23     num_frames = 120;
24     print("Capturing {0} frames".format(num_frames))
25
26     # Start time
27     start = time.time()
28     # Grab a few frames
29     for i in range(0, num_frames):
30         ret, frame = video.read()
31     # End time
32     end = time.time()
33
34     # Time elapsed
35     seconds = end - start
36     print("Time taken : {0} seconds".format(seconds))
37
38     # 计算 FPS, alculate frames per second
39     fps = num_frames / seconds;
40     print("Estimated frames per second : {0}".format(fps))
41
42     # 释放 video
43     video.release()
```
