

1 源代码

1.1 designer/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.
12
13
14 from PyQt5 import QtCore, QtGui, QtWidgets
15
16 class Ui_MainWindow(object):
17     def setupUi(self, MainWindow):
18         MainWindow.setObjectName("MainWindow")
19         MainWindow.resize(354, 366)
20         self.centralwidget = QtWidgets.QWidget(MainWindow)
21         self.centralwidget.setObjectName("centralwidget")
22         self.tabWidget =
23             ↪ QtWidgets.QTabWidget(self.centralwidget)
24         self.tabWidget.setGeometry(QtCore.QRect(10, 20, 321,
25             ↪ 251))
26         self.tabWidget.setObjectName("tabWidget")
27         self.tab = QtWidgets.QWidget()
28         self.tab.setObjectName("tab")
29         self.tab1_combobox = QtWidgets.QComboBox(self.tab)
30         self.tab1_combobox.setGeometry(QtCore.QRect(70, 0,
31             ↪ 171, 22))
32         self.tab1_combobox.setObjectName("tab1_combobox")
33         self.tab1_label = QtWidgets.QLabel(self.tab)
```

```
29         self.tab1_label.setGeometry(QRect(0, 30, 311,
        ↪ 191))
30         self.tab1_label.setObjectName("tab1_label")
31         self.tabWidget.addTab(self.tab, "")
32         self.tab_2 = QtWidgets.QWidget()
33         self.tab_2.setObjectName("tab_2")
34         self.tabWidget.addTab(self.tab_2, "")
35         self.pushButton =
        ↪ QtWidgets.QPushButton(self.centralwidget)
36         self.pushButton.setGeometry(QRect(230, 290, 75,
        ↪ 23))
37         self.pushButton.setObjectName("pushButton")
38         MainWindow.setCentralWidget(self.centralwidget)
39         self.menubar = QtWidgets.QMenuBar(MainWindow)
40         self.menubar.setGeometry(QRect(0, 0, 354, 23))
41         self.menubar.setObjectName("menubar")
42         MainWindow.setMenuBar(self.menubar)
43         self.statusbar = QtWidgets.QStatusBar(MainWindow)
44         self.statusbar.setObjectName("statusbar")
45         MainWindow.setStatusBar(self.statusbar)
46
47         self.retranslateUi(MainWindow)
48         QtCore.QMetaObject.connectSlotsByName(MainWindow)
49
50     def retranslateUi(self, MainWindow):
51         _translate = QtCore.QCoreApplication.translate
52         MainWindow.setWindowTitle(_translate("MainWindow",
        ↪ "MainWindow"))
53         self.tab1_label.setText(_translate("MainWindow",
        ↪ "TextLabel"))
54         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪
        ↪ .tab), _translate("MainWindow", "Tab
        ↪ 1"))
55         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪
        ↪ .tab_2), _translate("MainWindow", "Tab
        ↪ 2"))
56         self.pushButton.setText(_translate("MainWindow",
        ↪ "PushButton"))
57
```

```
58
59 if __name__ == "__main__":
60     import sys
61     app = QtWidgets.QApplication(sys.argv)
62     MainWindow = QtWidgets.QMainWindow()
63     ui = Ui_MainWindow()
64     ui.setupUi(MainWindow)
65     MainWindow.show()
66     sys.exit(app.exec_())
```

1.2 designer/child_setting.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>354</width>
10                <height>366</height>
11            </rect>
12        </property>
13        <property name="windowTitle">
14            <string>MainWindow</string>
15        </property>
16        <widget class="QWidget" name="centralwidget">
17            <widget class="QTabWidget" name="tabWidget">
18                <property name="geometry">
19                    <rect>
20                        <x>10</x>
21                        <y>20</y>
22                        <width>321</width>
23                        <height>251</height>
24                    </rect>
25                </property>
26                <widget class="QWidget" name="tab">
```

```
27     <attribute name="title">
28         <string>Tab 1</string>
29     </attribute>
30     <widget class="QComboBox" name="tab1_combobox">
31         <property name="geometry">
32             <rect>
33                 <x>70</x>
34                 <y>0</y>
35                 <width>171</width>
36                 <height>22</height>
37             </rect>
38         </property>
39     </widget>
40     <widget class="QLabel" name="tab1_label">
41         <property name="geometry">
42             <rect>
43                 <x>0</x>
44                 <y>30</y>
45                 <width>311</width>
46                 <height>191</height>
47             </rect>
48         </property>
49         <property name="text">
50             <string>TextLabel</string>
51         </property>
52     </widget>
53 </widget>
54 <widget class="QWidget" name="tab_2">
55     <attribute name="title">
56         <string>Tab 2</string>
57     </attribute>
58 </widget>
59 </widget>
60 <widget class="QPushButton" name="pushButton">
61     <property name="geometry">
62         <rect>
63             <x>230</x>
64             <y>290</y>
65             <width>75</width>
```

```

66         <height>23</height>
67     </rect>
68 </property>
69 <property name="text">
70     <string>PushButton</string>
71 </property>
72 </widget>
73 </widget>
74 <widget class="QMenuBar" name="menubar">
75     <property name="geometry">
76         <rect>
77             <x>0</x>
78             <y>0</y>
79             <width>354</width>
80             <height>23</height>
81         </rect>
82     </property>
83 </widget>
84 <widget class="QStatusBar" name="statusbar"/>
85 </widget>
86 <resources/>
87 <connections/>
88 </ui>

```

1.3 designer/child_test.py

```

1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file
4  ↳ 'child_test.ui'
5  #
6  # Created by: PyQt5 UI code generator 5.9.2
7  #
8  # WARNING! All changes made in this file will be lost!
9
10 from PyQt5 import QtCore, QtGui, QtWidgets
11
12 class Ui_Form(object):

```

```
12     def setupUi(self, Form):
13         Form.setObjectName("Form")
14         Form.resize(400, 310)
15         self.tabWidget = QtWidgets.QTabWidget(Form)
16         self.tabWidget.setGeometry(QtCore.QRect(10, 10, 381,
17             ↪ 261))
18         self.tabWidget.setObjectName("tabWidget")
19         self.tab = QtWidgets.QWidget()
20         self.tab.setObjectName("tab")
21         self.layoutWidget = QtWidgets.QWidget(self.tab)
22         self.layoutWidget.setGeometry(QtCore.QRect(0, 10, 371,
23             ↪ 211))
24         self.layoutWidget.setObjectName("layoutWidget")
25         self.horizontalLayout_2 =
26             ↪ QtWidgets.QHBoxLayout(self.layoutWidget)
27         self.horizontalLayout_2.setContentsMargins(0, 0, 0, 0)
28         self.horizontalLayout_2.setObjectName("horizontalLayout_2")
29         spacerItem = QtWidgets.QSpacerItem(40, 20,
30             ↪ QtWidgets.QSizePolicy.Expanding,
31             ↪ QtWidgets.QSizePolicy.Minimum)
32         self.horizontalLayout_2.addItem(spacerItem)
33         self.verticalLayout_2 = QtWidgets.QVBoxLayout()
34         self.verticalLayout_2.setObjectName("verticalLayout_2")
35         self.verticalLayout = QtWidgets.QVBoxLayout()
36         self.verticalLayout.setObjectName("verticalLayout")
37         self.horizontalLayout = QtWidgets.QHBoxLayout()
38         self.horizontalLayout.setObjectName("horizontalLayout")
39         spacerItem1 = QtWidgets.QSpacerItem(30, 20,
40             ↪ QtWidgets.QSizePolicy.Maximum,
41             ↪ QtWidgets.QSizePolicy.Minimum)
42         self.horizontalLayout.addItem(spacerItem1)
43         self.tab1_combobox =
44             ↪ QtWidgets.QComboBox(self.layoutWidget)
45         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy_
46             ↪ 1,
47             ↪ 1,
48             ↪ QtWidgets.QSizePolicy.MinimumExpanding,
49             ↪ QtWidgets.QSizePolicy.Fixed)
```

```
38         sizePolicy.setHorizontalStretch(0)
39         sizePolicy.setVerticalStretch(0)
40         sizePolicy.setHeightForWidth(self.tab1_combobox.sizeP
    ↪ olicy().hasHeightForWidth())
41         self.tab1_combobox.setSizePolicy(sizePolicy)
42         self.tab1_combobox.setObjectName("tab1_combobox")
43         self.horizontalLayout.addWidget(self.tab1_combobox)
44         spacerItem2 = QtWidgets.QSpacerItem(30, 20,
    ↪ QtWidgets.QSizePolicy.Maximum,
    ↪ QtWidgets.QSizePolicy.Minimum)
45         self.horizontalLayout.addItem(spacerItem2)
46         self.verticalLayout.addLayout(self.horizontalLayout)
47         spacerItem3 = QtWidgets.QSpacerItem(40, 20,
    ↪ QtWidgets.QSizePolicy.Expanding,
    ↪ QtWidgets.QSizePolicy.Minimum)
48         self.verticalLayout.addItem(spacerItem3)
49         self.verticalLayout_2.addLayout(self.verticalLayout)
50         self.tab1_label = QtWidgets.QLabel(self.layoutWidget)
51         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol
    ↪ icy.Ignored,
    ↪ QtWidgets.QSizePolicy.Fixed)
52         sizePolicy.setHorizontalStretch(0)
53         sizePolicy.setVerticalStretch(0)
54         sizePolicy.setHeightForWidth(self.tab1_label.sizePoli
    ↪ cy().hasHeightForWidth())
55         self.tab1_label.setSizePolicy(sizePolicy)
56         self.tab1_label.setMinimumSize(QtCore.QSize(256, 144))
57         self.tab1_label.setMaximumSize(QtCore.QSize(256, 144))
58         self.tab1_label.setAlignment(QtCore.Qt.AlignCenter)
59         self.tab1_label.setObjectName("tab1_label")
60         self.verticalLayout_2.addWidget(self.tab1_label)
61         self.horizontalLayout_2.addLayout(self.verticalLayout
    ↪ _2)
62         spacerItem4 = QtWidgets.QSpacerItem(40, 20,
    ↪ QtWidgets.QSizePolicy.Expanding,
    ↪ QtWidgets.QSizePolicy.Minimum)
63         self.horizontalLayout_2.addItem(spacerItem4)
64         self.tabWidget.addTab(self.tab, "")
65         self.tab_2 = QtWidgets.QWidget()
```

```
66     self.tab_2.setObjectName("tab_2")
67     self.layoutWidget_2 = QtWidgets.QWidget(self.tab_2)
68     self.layoutWidget_2.setGeometry(QtCore.QRect(0, 10,
        ↪ 371, 211))
69     self.layoutWidget_2.setObjectName("layoutWidget_2")
70     self.horizontalLayout_7 =
        ↪ QtWidgets.QHBoxLayout(self.layoutWidget_2)
71     self.horizontalLayout_7.setContentsMargins(0, 0, 0, 0)
72     self.horizontalLayout_7.setObjectName("horizontalLayo_
        ↪ ut_7")
73     spacerItem5 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
74     self.horizontalLayout_7.addItem(spacerItem5)
75     self.verticalLayout_5 = QtWidgets.QVBoxLayout()
76     self.verticalLayout_5.setObjectName("verticalLayout_5_
        ↪ ")
77     self.verticalLayout_6 = QtWidgets.QVBoxLayout()
78     self.verticalLayout_6.setObjectName("verticalLayout_6_
        ↪ ")
79     self.horizontalLayout_8 = QtWidgets.QHBoxLayout()
80     self.horizontalLayout_8.setObjectName("horizontalLayo_
        ↪ ut_8")
81     spacerItem6 = QtWidgets.QSpacerItem(30, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
82     self.horizontalLayout_8.addItem(spacerItem6)
83     self.tab1_combobox_3 =
        ↪ QtWidgets.QComboBox(self.layoutWidget_2)
84     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
        ↪  icy.MinimumExpanding,
        ↪ QtWidgets.QSizePolicy.Fixed)
85     sizePolicy.setHorizontalStretch(0)
86     sizePolicy.setVerticalStretch(0)
87     sizePolicy.setHeightForWidth(self.tab1_combobox_3.siz_
        ↪ ePolicy().hasHeightForWidth())
88     self.tab1_combobox_3.setSizePolicy(sizePolicy)
89     self.tab1_combobox_3.setObjectName("tab1_combobox_3")
```



```
90         self.horizontalLayout_8.addWidget(self.tab1_combobox_1
91         ↪ 3)
92     spacerItem7 = QtWidgets.QSpacerItem(30, 20,
93     ↪ QtWidgets.QSizePolicy.Maximum,
94     ↪ QtWidgets.QSizePolicy.Minimum)
95     self.horizontalLayout_8.addItem(spacerItem7)
96     self.verticalLayout_6.addLayout(self.horizontalLayout_8)
97     ↪ _8)
98     spacerItem8 = QtWidgets.QSpacerItem(40, 20,
99     ↪ QtWidgets.QSizePolicy.Expanding,
100    ↪ QtWidgets.QSizePolicy.Minimum)
101    self.verticalLayout_6.addItem(spacerItem8)
102    self.verticalLayout_5.addLayout(self.verticalLayout_6)
103    self.tab1_label_3 =
104    ↪ QtWidgets.QLabel(self.layoutWidget_2)
105    sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy_
106    ↪ icky.Ignored,
107    ↪ QtWidgets.QSizePolicy.Fixed)
108    sizePolicy.setHorizontalStretch(0)
109    sizePolicy.setVerticalStretch(0)
110    sizePolicy.setHeightForWidth(self.tab1_label_3.sizePo_
111    ↪ licy().hasHeightForWidth())
112    self.tab1_label_3.setSizePolicy(sizePolicy)
113    self.tab1_label_3.setMinimumSize(QtCore.QSize(256,
114    ↪ 144))
115    self.tab1_label_3.setMaximumSize(QtCore.QSize(256,
116    ↪ 144))
117    self.tab1_label_3.setAlignment(QtCore.Qt.AlignCenter)
118    self.tab1_label_3.setObjectName("tab1_label_3")
119    self.verticalLayout_5.addWidget(self.tab1_label_3)
120    self.horizontalLayout_7.addLayout(self.verticalLayout_
121    ↪ 5)
122    spacerItem9 = QtWidgets.QSpacerItem(40, 20,
123    ↪ QtWidgets.QSizePolicy.Expanding,
124    ↪ QtWidgets.QSizePolicy.Minimum)
125    self.horizontalLayout_7.addItem(spacerItem9)
126    self.tabWidget.addTab(self.tab_2, "")
127    self.tab_3 = QtWidgets.QWidget()
128    self.tab_3.setObjectName("tab_3")
```

```

114         self.layoutWidget_3 = QtWidgets.QWidget(self.tab_3)
115         self.layoutWidget_3.setGeometry(QtCore.QRect(0, 10,
116             ↪ 371, 211))
117         self.layoutWidget_3.setObjectName("layoutWidget_3")
118         self.horizontalLayout_4 =
119             ↪ QtWidgets.QHBoxLayout(self.layoutWidget_3)
120         self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
121         self.horizontalLayout_4.setObjectName("horizontalLayo_
122             ↪ ut_4")
123         spacerItem10 = QtWidgets.QSpacerItem(40, 20,
124             ↪ QtWidgets.QSizePolicy.Expanding,
125             ↪ QtWidgets.QSizePolicy.Minimum)
126         self.horizontalLayout_4.addItem(spacerItem10)
127         self.verticalLayout_7 = QtWidgets.QVBoxLayout()
128         self.verticalLayout_7.setObjectName("verticalLayout_7_
129             ↪ ")
130         self.verticalLayout_8 = QtWidgets.QVBoxLayout()
131         self.verticalLayout_8.setObjectName("verticalLayout_8_
132             ↪ ")
133         self.horizontalLayout_9 = QtWidgets.QHBoxLayout()
134         self.horizontalLayout_9.setObjectName("horizontalLayo_
135             ↪ ut_9")
136         spacerItem11 = QtWidgets.QSpacerItem(30, 20,
137             ↪ QtWidgets.QSizePolicy.Maximum,
138             ↪ QtWidgets.QSizePolicy.Minimum)
139         self.horizontalLayout_9.addItem(spacerItem11)
140         self.tab1_combobox_4 =
141             ↪ QtWidgets.QComboBox(self.layoutWidget_3)
142         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
143             ↪ icky.MinimumExpanding,
144             ↪ QtWidgets.QSizePolicy.Fixed)
145         sizePolicy.setHorizontalStretch(0)
146         sizePolicy.setVerticalStretch(0)
147         sizePolicy.setHeightForWidth(self.tab1_combobox_4.siz_
148             ↪ ePolicy().hasHeightForWidth())
149         self.tab1_combobox_4.setSizePolicy(sizePolicy)
150         self.tab1_combobox_4.setObjectName("tab1_combobox_4")
151         self.horizontalLayout_9.addWidget(self.tab1_combobox_
152             ↪ 4)

```

```
138         spacerItem12 = QtWidgets.QSpacerItem(30, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
139     self.horizontalLayout_9.addItem(spacerItem12)
140     self.verticalLayout_8.addLayout(self.horizontalLayout_
        ↪ _9)
141     spacerItem13 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
142     self.verticalLayout_8.addItem(spacerItem13)
143     self.verticalLayout_7.addLayout(self.verticalLayout_8)
144     self.tab1_label_4 =
        ↪ QtWidgets.QLabel(self.layoutWidget_3)
145     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol
        ↪ icy.Ignored,
        ↪ QtWidgets.QSizePolicy.Fixed)
146     sizePolicy.setHorizontalStretch(0)
147     sizePolicy.setVerticalStretch(0)
148     sizePolicy.setHeightForWidth(self.tab1_label_4.sizePo
        ↪ licy().hasHeightForWidth())
149     self.tab1_label_4.setSizePolicy(sizePolicy)
150     self.tab1_label_4.setMinimumSize(QtCore.QSize(256,
        ↪ 144))
151     self.tab1_label_4.setMaximumSize(QtCore.QSize(256,
        ↪ 144))
152     self.tab1_label_4.setAlignment(QtCore.Qt.AlignCenter)
153     self.tab1_label_4.setObjectName("tab1_label_4")
154     self.verticalLayout_7.addWidget(self.tab1_label_4)
155     self.horizontalLayout_4.addLayout(self.verticalLayout_
        ↪ _7)
156     spacerItem14 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
157     self.horizontalLayout_4.addItem(spacerItem14)
158     self.tabWidget.addTab(self.tab_3, "")
159     self.tab_4 = QtWidgets.QWidget()
160     self.tab_4.setObjectName("tab_4")
161     self.layoutWidget1 = QtWidgets.QWidget(self.tab_4)
```

```
162         self.layoutWidget1.setGeometry(QRect(0, 10,
        ↪ 371, 211))
163         self.layoutWidget1.setObjectName("layoutWidget1")
164         self.horizontalLayout_13 =
        ↪ QtWidgets.QHBoxLayout(self.layoutWidget1)
165         self.horizontalLayout_13.setContentsMargins(0, 0, 0,
        ↪ 0)
166         self.horizontalLayout_13.setObjectName("horizontalLay_
        ↪ out_13")
167         spacerItem15 = QtWidgets.QSpacerItem(40, 40,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
168         self.horizontalLayout_13.addItem(spacerItem15)
169         self.verticalLayout_13 = QtWidgets.QVBoxLayout()
170         self.verticalLayout_13.setObjectName("verticalLayout_
        ↪ 13")
171         self.verticalLayout_14 = QtWidgets.QVBoxLayout()
172         self.verticalLayout_14.setObjectName("verticalLayout_
        ↪ 14")
173         self.horizontalLayout_14 = QtWidgets.QHBoxLayout()
174         self.horizontalLayout_14.setObjectName("horizontalLay_
        ↪ out_14")
175         spacerItem16 = QtWidgets.QSpacerItem(31, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
176         self.horizontalLayout_14.addItem(spacerItem16)
177         self.tab1_combobox_7 =
        ↪ QtWidgets.QComboBox(self.layoutWidget1)
178         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
        ↪ icky.MinimumExpanding,
        ↪ QtWidgets.QSizePolicy.Fixed)
179         sizePolicy.setHorizontalStretch(0)
180         sizePolicy.setVerticalStretch(0)
181         sizePolicy.setHeightForWidth(self.tab1_combobox_7.siz_
        ↪ ePolicy().hasHeightForWidth())
182         self.tab1_combobox_7.setSizePolicy(sizePolicy)
183         self.tab1_combobox_7.setMinimumSize(QtCore.QSize(160,
        ↪ 0))
184         self.tab1_combobox_7.setObjectName("tab1_combobox_7")
```

```
185         self.horizontalLayout_14.addWidget(self.tab1_combobox_1
        ↪ _7)
186         spacerItem17 = QtWidgets.QSpacerItem(31, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
187         self.horizontalLayout_14.addItem(spacerItem17)
188         self.verticalLayout_14.addLayout(self.horizontalLayout_1
        ↪ t_14)
189         spacerItem18 = QtWidgets.QSpacerItem(256, 55,
        ↪ QtWidgets.QSizePolicy.Minimum,
        ↪ QtWidgets.QSizePolicy.Minimum)
190         self.verticalLayout_14.addItem(spacerItem18)
191         self.verticalLayout_13.addLayout(self.verticalLayout_1
        ↪ 14)
192         self.progressBar =
        ↪ QtWidgets.QProgressBar(self.layoutWidget1)
193         self.progressBar.setProperty("value", 0)
194         self.progressBar.setTextVisible(False)
195         self.progressBar.setObjectName("progressBar")
196         self.verticalLayout_13.addWidget(self.progressBar)
197         spacerItem19 = QtWidgets.QSpacerItem(40, 60,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
198         self.verticalLayout_13.addItem(spacerItem19)
199         self.horizontalLayout_13.addLayout(self.verticalLayout_1
        ↪ t_13)
200         spacerItem20 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
201         self.horizontalLayout_13.addItem(spacerItem20)
202         self.tabWidget.addTab(self.tab_4, "")
203         self.OK = QtWidgets.QPushButton(Form)
204         self.OK.setGeometry(QtCore.QRect(260, 280, 111, 23))
205         self.OK.setObjectName("OK")
206
207         self.retranslateUi(Form)
208         self.tabWidget.setCurrentIndex(0)
209         QtCore.QMetaObject.connectSlotsByName(Form)
210
```

```

211     def retranslateUi(self, Form):
212         _translate = QtCore.QCoreApplication.translate
213         Form.setWindowTitle(_translate("Form", "Form"))
214         self.tab1_label.setText(_translate("Form",
215             ↪ "TextLabel"))
216         self.tabWidget.setTabText(self.tabWidget.indexOf(self
217             ↪ .tab), _translate("Form",
218             ↪ "行为"))
219         self.tab1_label_3.setText(_translate("Form",
220             ↪ "TextLabel"))
221         self.tabWidget.setTabText(self.tabWidget.indexOf(self
222             ↪ .tab_2), _translate("Form",
223             ↪ "面部"))
224         self.tab1_label_4.setText(_translate("Form",
225             ↪ "TextLabel"))
226         self.tabWidget.setTabText(self.tabWidget.indexOf(self
227             ↪ .tab_3), _translate("Form",
228             ↪ "眼睛"))
229         self.tabWidget.setTabText(self.tabWidget.indexOf(self
230             ↪ .tab_4), _translate("Form",
231             ↪ "麦克风"))
232         self.OK.setText(_translate("Form", " 确认"))

```

1.4 designer/child_test.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>400</width>
10                 <height>310</height>
11             </rect>
12         </property>

```

```
13 <property name="windowTitle">
14   <string>Form</string>
15 </property>
16 <widget class="QTabWidget" name="tabWidget">
17   <property name="geometry">
18     <rect>
19       <x>10</x>
20       <y>10</y>
21       <width>381</width>
22       <height>261</height>
23     </rect>
24   </property>
25   <property name="currentIndex">
26     <number>0</number>
27   </property>
28   <widget class="QWidget" name="tab">
29     <attribute name="title">
30       <string>行为</string>
31     </attribute>
32     <widget class="QWidget" name="layoutWidget">
33       <property name="geometry">
34         <rect>
35           <x>0</x>
36           <y>10</y>
37           <width>371</width>
38           <height>211</height>
39         </rect>
40       </property>
41       <layout class="QHBoxLayout" name="horizontalLayout_2">
42         <item>
43           <spacer name="horizontalSpacer_10">
44             <property name="orientation">
45               <enum>Qt::Horizontal</enum>
46             </property>
47             <property name="sizeHint" stdset="0">
48               <size>
49                 <width>40</width>
50                 <height>20</height>
51               </size>
```

```

52         </property>
53     </spacer>
54 </item>
55 <item>
56     <layout class="QVBoxLayout" name="verticalLayout_2">
57         <item>
58             <layout class="QVBoxLayout" name="verticalLayout">
59                 <item>
60                     <layout class="QHBoxLayout"
61                     ↪ name="horizontalLayout">
62                         <item>
63                             <spacer name="horizontalSpacer">
64                                 <property name="orientation">
65                                     <enum>Qt::Horizontal</enum>
66                                 </property>
67                                 <property name="sizeType">
68                                     <enum>QSizePolicy::Maximum</enum>
69                                 </property>
70                                 <property name="sizeHint" stdset="0">
71                                     <size>
72                                         <width>30</width>
73                                         <height>20</height>
74                                     </size>
75                                 </property>
76                             </spacer>
77                         </item>
78                         <item>
79                             <widget class="QComboBox" name="tab1_combobox">
80                                 <property name="sizePolicy">
81                                     <sizepolicy hsizeType="MinimumExpanding"
82                                     ↪ vsizeType="Fixed">
83                                         <horstretch>0</horstretch>
84                                         <verstretch>0</verstretch>
85                                     </sizepolicy>
86                                 </property>
87                             </widget>
88                         </item>
89                         <item>
90                             <spacer name="horizontalSpacer_2">

```



```
89         <property name="orientation">
90             <enum>Qt::Horizontal</enum>
91         </property>
92         <property name="sizeType">
93             <enum>QSizePolicy::Maximum</enum>
94         </property>
95         <property name="sizeHint" stdset="0">
96             <size>
97                 <width>30</width>
98                 <height>20</height>
99             </size>
100         </property>
101     </spacer>
102 </item>
103 </layout>
104 </item>
105 <item>
106     <spacer name="horizontalSpacer_9">
107         <property name="orientation">
108             <enum>Qt::Horizontal</enum>
109         </property>
110         <property name="sizeHint" stdset="0">
111             <size>
112                 <width>40</width>
113                 <height>20</height>
114             </size>
115         </property>
116     </spacer>
117 </item>
118 </layout>
119 </item>
120 <item>
121     <widget class="QLabel" name="tab1_label">
122         <property name="sizePolicy">
123             <sizepolicy hsiptype="Ignored" vsizetype="Fixed">
124                 <horstretch>0</horstretch>
125                 <verstretch>0</verstretch>
126             </sizepolicy>
127         </property>
```

```
128         <property name="minimumSize">
129             <size>
130                 <width>256</width>
131                 <height>144</height>
132             </size>
133         </property>
134         <property name="maximumSize">
135             <size>
136                 <width>256</width>
137                 <height>144</height>
138             </size>
139         </property>
140         <property name="text">
141             <string>TextLabel</string>
142         </property>
143         <property name="alignment">
144             <set>Qt::AlignCenter</set>
145         </property>
146     </widget>
147 </item>
148 </layout>
149 </item>
150 <item>
151     <spacer name="horizontalSpacer_11">
152         <property name="orientation">
153             <enum>Qt::Horizontal</enum>
154         </property>
155         <property name="sizeHint" stdset="0">
156             <size>
157                 <width>40</width>
158                 <height>20</height>
159             </size>
160         </property>
161     </spacer>
162 </item>
163 </layout>
164 </widget>
165 </widget>
166 <widget class="QWidget" name="tab_2">
```

[illegible]

```

205         <enum>QSizePolicy::Maximum</enum>
206     </property>
207     <property name="sizeHint" stdset="0">
208         <size>
209             <width>30</width>
210             <height>20</height>
211         </size>
212     </property>
213 </spacer>
214 </item>
215 <item>
216     <widget class="QComboBox" name="tab1_combobox_3">
217         <property name="sizePolicy">
218             <sizepolicy hsizeType="MinimumExpanding"
219                 ↪ vsizeType="Fixed">
220                 <horstretch>0</horstretch>
221                 <verstretch>0</verstretch>
222             </sizepolicy>
223         </property>
224     </widget>
225 </item>
226 <item>
227     <spacer name="horizontalSpacer_17">
228         <property name="orientation">
229             <enum>Qt::Horizontal</enum>
230         </property>
231         <property name="sizeType">
232             <enum>QSizePolicy::Maximum</enum>
233         </property>
234         <property name="sizeHint" stdset="0">
235             <size>
236                 <width>30</width>
237                 <height>20</height>
238             </size>
239         </property>
240     </spacer>
241 </item>
242 </layout>
</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 hstretch="Ignored" vstretch="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">
```



```

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>
339                             <spacer name="horizontalSpacer_5">
340                                 <property name="orientation">
341                                     <enum>Qt::Horizontal</enum>
342                                 </property>
343                                 <property name="sizeType">
344                                     <enum>QSizePolicy::Maximum</enum>
345                                 </property>
346                                 <property name="sizeHint" stdset="0">
347                                     <size>
348                                         <width>30</width>
349                                         <height>20</height>
350                                     </size>
351                                 </property>
352                             </spacer>
353                         </item>
354                         <item>
355                             <widget class="QComboBox" name="tab1_combobox_4">
356                                 <property name="sizePolicy">
357                                     <sizepolicy hsizeType="MinimumExpanding"
358                                         ↪ vsizetype="Fixed">
359                                     <horstretch>0</horstretch>

```

```
358         <verstretch>0</verstretch>
359     </sizepolicy>
360 </property>
361 </widget>
362 </item>
363 <item>
364     <spacer name="horizontalSpacer_6">
365         <property name="orientation">
366             <enum>Qt::Horizontal</enum>
367         </property>
368         <property name="sizeType">
369             <enum>QSizePolicy::Maximum</enum>
370         </property>
371         <property name="sizeHint" stdset="0">
372             <size>
373                 <width>30</width>
374                 <height>20</height>
375             </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 hsizeType="Ignored" vsizeType="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"
    ↳ name="horizontalLayout_14">
475 <item>
476 <spacer name="horizontalSpacer_32">
477 <property name="orientation">
478 <enum>Qt::Horizontal</enum>
479 </property>
480 <property name="sizeType">
481 <enum>QSizePolicy::Maximum</enum>
482 </property>
483 <property name="sizeHint" stdset="0">
484 <size>
485 <width>31</width>
486 <height>20</height>
487 </size>
488 </property>
489 </spacer>
490 </item>
491 <item>
492 <widget class="QComboBox" name="tab1_combobox_7">
493 <property name="sizePolicy">
494 <sizepolicy hsiptype="MinimumExpanding"
    ↳ vsizetype="Fixed">
495 <horstretch>0</horstretch>
496 <verstretch>0</verstretch>
497 </sizepolicy>
498 </property>
499 <property name="minimumSize">
500 <size>
501 <width>160</width>
502 <height>0</height>
503 </size>
504 </property>
505 </widget>
506 </item>
507 <item>
508 <spacer name="horizontalSpacer_33">
509 <property name="orientation">
510 <enum>Qt::Horizontal</enum>
```

```
511         </property>
512         <property name="sizeType">
513             <enum>QSizePolicy::Maximum</enum>
514         </property>
515         <property name="sizeHint" stdset="0">
516             <size>
517                 <width>31</width>
518                 <height>20</height>
519             </size>
520         </property>
521     </spacer>
522 </item>
523 </layout>
524 </item>
525 <item>
526     <spacer name="horizontalSpacer_34">
527         <property name="orientation">
528             <enum>Qt::Horizontal</enum>
529         </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>
```

1.5 designer/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 class Ui_MainWindow(object):
18     def setupUi(self, MainWindow):
19         MainWindow.setObjectName("MainWindow")
20         MainWindow.resize(1048, 1025)
21         icon = QtGui.QIcon()
22         icon.addPixmap(QtGui.QPixmap("../icon.ico"),
23             QtGui.QIcon.Normal, QtGui.QIcon.Off)
```

```
20     MainWindow.setWindowIcon(icon)
21     self.centralwidget = QtWidgets.QWidget(MainWindow)
22     self.centralwidget.setObjectName("centralwidget")
23     self.verticalLayout_4 =
24         ↳ QtWidgets.QVBoxLayout(self.centralwidget)
25     self.verticalLayout_4.setObjectName("verticalLayout_4_
26         ↳ ")
27     self.h_img_layout = QtWidgets.QHBoxLayout()
28     self.h_img_layout.setObjectName("h_img_layout")
29     self.original_img_v_layout = QtWidgets.QVBoxLayout()
30     self.original_img_v_layout.setObjectName("original_img_v_
31         ↳ _layout")
32     spacerItem = QtWidgets.QSpacerItem(20, 40,
33         ↳ QtWidgets.QSizePolicy.Minimum,
34         ↳ QtWidgets.QSizePolicy.Expanding)
35     self.original_img_v_layout.addItem(spacerItem)
36     self.original_name_label =
37         ↳ QtWidgets.QLabel(self.centralwidget)
38     self.original_name_label.setScaledContents(False)
39     self.original_name_label.setAlignment(QtCore.Qt.AlignCe
40         ↳ nter)
41     self.original_name_label.setObjectName("original_name_l_
42         ↳ bl")
43     self.original_img_v_layout.addWidget(self.original_name_
44         ↳ label)
45     spacerItem1 = QtWidgets.QSpacerItem(20, 40,
46         ↳ QtWidgets.QSizePolicy.Minimum,
47         ↳ QtWidgets.QSizePolicy.Expanding)
48     self.original_img_v_layout.addItem(spacerItem1)
49     self.original_frame_label =
50         ↳ QtWidgets.QLabel(self.centralwidget)
51     self.original_frame_label.setMinimumSize(QtCore.QSize(5
52         ↳ 00,
53         ↳ 300))
54     self.original_frame_label.setObjectName("original_frame_
55         ↳ _label")
56     self.original_img_v_layout.addWidget(self.original_frame_
57         ↳ label)
```

```
42     self.selecamera1 =
43         ↪ QtWidgets.QComboBox(self.centralwidget)
44     self.selecamera1.setObjectName("selecamera1")
45     self.original_img_v_layout.addWidget(self.selecamera1)
46     spacerItem2 = QtWidgets.QSpacerItem(20, 40,
47         ↪ QtWidgets.QSizePolicy.Minimum,
48         ↪ QtWidgets.QSizePolicy.Expanding)
49     self.original_img_v_layout.addItem(spacerItem2)
50     self.h_img_layout.addLayout(self.original_img_v_layout)
51     self.processed_img_v_layout = QtWidgets.QVBoxLayout()
52     self.processed_img_v_layout.setObjectName("processed_img_v_
53         ↪ _layout")
54     spacerItem3 = QtWidgets.QSpacerItem(20, 40,
55         ↪ QtWidgets.QSizePolicy.Minimum,
56         ↪ QtWidgets.QSizePolicy.Expanding)
57     self.processed_img_v_layout.addItem(spacerItem3)
58     self.processed_name_label =
59         ↪ QtWidgets.QLabel(self.centralwidget)
60     self.processed_name_label.setAlignment(QtCore.Qt.AlignC_
61         ↪ enter)
62     self.processed_name_label.setObjectName("processed_name_
63         ↪ _label")
64     self.processed_img_v_layout.addWidget(self.processed_nam_
65         ↪ e_label)
66     spacerItem4 = QtWidgets.QSpacerItem(20, 40,
67         ↪ QtWidgets.QSizePolicy.Minimum,
68         ↪ QtWidgets.QSizePolicy.Expanding)
69     self.processed_img_v_layout.addItem(spacerItem4)
70     self.processed_frame_label =
71         ↪ QtWidgets.QLabel(self.centralwidget)
72     self.processed_frame_label.setMinimumSize(QtCore.QSize(
73         ↪ 500,
74         ↪ 300))
75     self.processed_frame_label.setBaseSize(QtCore.QSize(0,
76         ↪ 0))
77     self.processed_frame_label.setText("")
78     self.processed_frame_label.setObjectName("processed_fra_
79         ↪ me_label")
```



```

63         self.processed_img_v_layout.addWidget(self.processed_frame_label,
        ↪ me_label)
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_label = QtWidgets.QLabel(self.centralwidget)
79         self.eye_label.setMinimumSize(QtCore.QSize(500, 300))
80         self.eye_label.setText("")
81         self.eye_label.setObjectName("eye_label")
82         self.verticalLayout_2.addWidget(self.eye_label)
83         self.comboBox_2 =
        ↪ QtWidgets.QComboBox(self.centralwidget)
84         self.comboBox_2.setObjectName("comboBox_2")
85         self.verticalLayout_2.addWidget(self.comboBox_2)
86         self.horizontalLayout.addLayout(self.verticalLayout_2)
87         self.verticalLayout = QtWidgets.QVBoxLayout()
88         self.verticalLayout.setObjectName("verticalLayout")
89         self.label_2 = QtWidgets.QLabel(self.centralwidget)
90         self.label_2.setObjectName("label_2")
91         self.verticalLayout.addWidget(self.label_2)
92         self.voice_label = QtWidgets.QLabel(self.centralwidget)
93         self.voice_label.setMinimumSize(QtCore.QSize(500, 300))
94         self.voice_label.setText("")

```

```
95         self.voice_lbl.setObjectName("voice_lbl")
96         self.verticalLayout.addWidget(self.voice_lbl)
97         self.comboBox =
98             ↳ QtWidgets.QComboBox(self.centralwidget)
99         self.comboBox.setObjectName("comboBox")
100        self.verticalLayout.addWidget(self.comboBox)
101        self.horizontalLayout.addLayout(self.verticalLayout)
102        self.verticalLayout_4.addLayout(self.horizontalLayout)
103        self.h_btn_lay = QtWidgets.QHBoxLayout()
104        self.h_btn_lay.setObjectName("h_btn_lay")
105        self.initButton =
106            ↳ QtWidgets.QPushButton(self.centralwidget)
107        self.initButton.setObjectName("initButton")
108        self.h_btn_lay.addWidget(self.initButton)
109        self.pushButton =
110            ↳ QtWidgets.QPushButton(self.centralwidget)
111        self.pushButton.setObjectName("pushButton")
112        self.h_btn_lay.addWidget(self.pushButton)
113        spacerItem6 = QtWidgets.QSpacerItem(40, 20,
114            ↳ QtWidgets.QSizePolicy.Minimum,
115            ↳ QtWidgets.QSizePolicy.Expanding)
116        self.h_btn_lay.addItem(spacerItem6)
117        self.label = QtWidgets.QLabel(self.centralwidget)
118        self.label.setObjectName("label")
119        self.h_btn_lay.addWidget(self.label)
120        self.lineEdit =
121            ↳ QtWidgets.QLineEdit(self.centralwidget)
122        self.lineEdit.setObjectName("lineEdit")
123        self.h_btn_lay.addWidget(self.lineEdit)
124        self.startButton =
125            ↳ QtWidgets.QPushButton(self.centralwidget)
126        self.startButton.setObjectName("startButton")
127        self.h_btn_lay.addWidget(self.startButton)
128        self.endButton =
129            ↳ QtWidgets.QPushButton(self.centralwidget)
130        self.endButton.setObjectName("endButton")
131        self.h_btn_lay.addWidget(self.endButton)
132        self.shotButton =
133            ↳ QtWidgets.QPushButton(self.centralwidget)
```

```
125         self.pushButton.setObjectName("shotButton")
126         self.h_btn_lay.addWidget(self.pushButton)
127         self.verticalLayout_4.addLayout(self.h_btn_lay)
128         self.v_filters_lay = QtWidgets.QVBoxLayout()
129         self.v_filters_lay.setObjectName("v_filters_lay")
130         self.verticalLayout_4.addLayout(self.v_filters_lay)
131         MainWindow.setCentralWidget(self.centralwidget)
132         self.menubar = QtWidgets.QMenuBar(MainWindow)
133         self.menubar.setGeometry(QtCore.QRect(0, 0, 1048, 37))
134         self.menubar.setObjectName("menubar")
135         self.menuHelp = QtWidgets.QMenu(self.menubar)
136         self.menuHelp.setObjectName("menuHelp")
137         self.menu = QtWidgets.QMenu(self.menubar)
138         self.menu.setObjectName("menu")
139         self.menu_2 = QtWidgets.QMenu(self.menubar)
140         self.menu_2.setObjectName("menu_2")
141         MainWindow.setMenuBar(self.menubar)
142         self.statusbar = QtWidgets.QStatusBar(MainWindow)
143         self.statusbar.setMinimumSize(QtCore.QSize(500, 30))
144         self.statusbar.setObjectName("statusbar")
145         MainWindow.setStatusBar(self.statusbar)
146         self.actionOpen_image = QtWidgets.QAction(MainWindow)
147         self.actionOpen_image.setObjectName("actionOpen_image_1")
148         self.actionSave_original_image =
149             QtWidgets.QAction(MainWindow)
150         self.actionSave_original_image.setObjectName("actionSave_Original_image")
151         self.actionSave_processed_image =
152             QtWidgets.QAction(MainWindow)
153         self.actionSave_processed_image.setObjectName("actionSave_Processed_image")
154         self.actionExit = QtWidgets.QAction(MainWindow)
155         self.actionExit.setObjectName("actionExit")
156         self.actionLicense = QtWidgets.QAction(MainWindow)
157         self.actionLicense.setObjectName("actionLicense")
158         self.actionAbout = QtWidgets.QAction(MainWindow)
159         self.actionAbout.setObjectName("actionAbout")
160         self.actionmanual = QtWidgets.QAction(MainWindow)
```

```

159         self.actionmanual.setObjectName("actionmanual")
160         self.action12 = QtWidgets.QAction(MainWindow)
161         self.action12.setObjectName("action12")
162         self.actionse = QtWidgets.QAction(MainWindow)
163         self.actionse.setObjectName("actionse")
164         self.menuHelp.addAction(self.actionmanual)
165         self.menuHelp.addAction(self.actionAbout)
166         self.menu_2.addAction(self.action12)
167         self.menu_2.addAction(self.actionse)
168         self.menubar.addAction(self.menu.menuAction())
169         self.menubar.addAction(self.menu_2.menuAction())
170         self.menubar.addAction(self.menuHelp.menuAction())
171
172         self.retranslateUi(MainWindow)
173         QtCore.QMetaObject.connectSlotsByName(MainWindow)
174
175     def retranslateUi(self, MainWindow):
176         _translate = QtCore.QCoreApplication.translate
177         MainWindow.setWindowTitle(_translate("MainWindow",
178         ↪ "面部表型特征提取系统---山东省精神卫生中心"))
179         self.original_name_lbl.setText(_translate("MainWindow",
180         ↪ " ",
181         ↪ "原始图像"))
182         self.original_frame_lbl.setText(_translate("MainWindo",
183         ↪ "w",
184         ↪ "<html><head/><body><p><br/></p></body></html>"))
185         self.processed_name_lbl.setText(_translate("MainWindo",
186         ↪ "w",
187         ↪ "面部特征"))
188         self.label_3.setText(_translate("MainWindow",
189         ↪ "标记图像"))
190         self.label_2.setText(_translate("MainWindow",
191         ↪ "情绪识别"))
192         self.initButton.setText(_translate("MainWindow",
193         ↪ "检测视频设备"))
194         self.pushButton.setText(_translate("MainWindow",
195         ↪ "停用视频设备"))
196         self.label.setText(_translate("MainWindow", "样本编号:
197         ↪ "))

```

```
186         self.startButton.setText(_translate("MainWindow",
187             ↪ "开始提取"))
188         self.endButton.setText(_translate("MainWindow",
189             ↪ "结束提取"))
190         self.startButton.setText(_translate("MainWindow",
191             ↪ "视频截图"))
192         self.menuHelp.setTitle(_translate("MainWindow",
193             ↪ "帮助"))
194         self.menu.setTitle(_translate("MainWindow", " 文件"))
195         self.menu_2.setTitle(_translate("MainWindow", "设置"))
196         self.actionOpen_image.setText(_translate("MainWindow",
197             ↪ "Open image"))
198         self.actionSave_original_image.setText(_translate("Ma
199             ↪ inWindow", "Save original
200             ↪ image"))
201         self.actionSave_processed_image.setText(_translate("M
202             ↪ ainWindow", "Save processed
203             ↪ image"))
204         self.actionExit.setText(_translate("MainWindow",
205             ↪ "Exit"))
206         self.actionLicense.setText(_translate("MainWindow",
207             ↪ "License"))
208         self.actionAbout.setText(_translate("MainWindow",
209             ↪ "关于"))
210         self.actionmanual.setText(_translate("MainWindow",
211             ↪ "操作说明"))
212         self.action12.setText(_translate("MainWindow",
213             ↪ "设置视频与音频设备"))
214         self.actionse.setText(_translate("MainWindow",
215             ↪ "设置文件保存路径"))
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```
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_())
```

1.6 designer/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;/htm_
        ↪      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: 

```

```
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>
```

1.7 designer/test.py

```
1  # -*- coding: utf-8 -*-
2
3  # Form implementation generated from reading ui file 'test.ui'
4  #
5  # Created by: PyQt5 UI code generator 5.9.2
6  #
7  # WARNING! All changes made in this file will be lost!
8
9  from PyQt5 import QtCore, QtGui, QtWidgets
10
11 class Ui_Form(object):
12     def setupUi(self, Form):
13         Form.setObjectName("Form")
14         Form.resize(400, 300)
```

```
15     self.tabWidget = QtWidgets.QTabWidget(Form)
16     self.tabWidget.setGeometry(QtCore.QRect(10, 10, 371,
    ↪ 261))
17     self.tabWidget.setObjectName("tabWidget")
18     self.tab = QtWidgets.QWidget()
19     self.tab.setObjectName("tab")
20     self.splitter = QtWidgets.QSplitter(self.tab)
21     self.splitter.setGeometry(QtCore.QRect(0, 10, 361,
    ↪ 221))
22     self.splitter.setOrientation(QtCore.Qt.Vertical)
23     self.splitter.setObjectName("splitter")
24     self.widget = QtWidgets.QWidget(self.splitter)
25     self.widget.setObjectName("widget")
26     self.horizontalLayout =
    ↪ QtWidgets.QHBoxLayout(self.widget)
27     self.horizontalLayout.setSizeConstraint(QtWidgets.QLa_
    ↪ yout.SetMinAndMaxSize)
28     self.horizontalLayout.setContentsMargins(0, 0, 0, 0)
29     self.horizontalLayout.setObjectName("horizontalLayout_
    ↪ ")
30     spacerItem = QtWidgets.QSpacerItem(90, 20,
    ↪ QtWidgets.QSizePolicy.Maximum,
    ↪ QtWidgets.QSizePolicy.Minimum)
31     self.horizontalLayout.addItem(spacerItem)
32     self.tab1_combobox = QtWidgets.QComboBox(self.widget)
33     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
    ↪ ICY.MinimumExpanding,
    ↪ QtWidgets.QSizePolicy.Fixed)
34     sizePolicy.setHorizontalStretch(0)
35     sizePolicy.setVerticalStretch(0)
36     sizePolicy.setHeightForWidth(self.tab1_combobox.sizeP_
    ↪ olicy().hasHeightForWidth())
37     self.tab1_combobox.setSizePolicy(sizePolicy)
38     self.tab1_combobox.setObjectName("tab1_combobox")
39     self.horizontalLayout.addWidget(self.tab1_combobox)
40     spacerItem1 = QtWidgets.QSpacerItem(90, 20,
    ↪ QtWidgets.QSizePolicy.Maximum,
    ↪ QtWidgets.QSizePolicy.Minimum)
41     self.horizontalLayout.addItem(spacerItem1)
```



```
42     self.tab1_label = QtWidgets.QLabel(self.splitter)
43     self.tab1_label.setObjectName("tab1_label")
44     self.tabWidget.addTab(self.tab, "")
45     self.tab_2 = QtWidgets.QWidget()
46     self.tab_2.setObjectName("tab_2")
47     self.splitter_2 = QtWidgets.QSplitter(self.tab_2)
48     self.splitter_2.setGeometry(QtCore.QRect(0, 10, 361,
49         ↪ 221))
49     self.splitter_2.setOrientation(QtCore.Qt.Vertical)
50     self.splitter_2.setObjectName("splitter_2")
51     self.widget1 = QtWidgets.QWidget(self.splitter_2)
52     self.widget1.setObjectName("widget1")
53     self.horizontalLayout_3 =
54         ↪ QtWidgets.QHBoxLayout(self.widget1)
54     self.horizontalLayout_3.setContentsMargins(0, 0, 0, 0)
55     self.horizontalLayout_3.setObjectName("horizontalLayo_
56         ↪ ut_3")
56     spacerItem2 = QtWidgets.QSpacerItem(90, 20,
57         ↪ QtWidgets.QSizePolicy.Maximum,
58         ↪ QtWidgets.QSizePolicy.Minimum)
57     self.horizontalLayout_3.addItem(spacerItem2)
58     self.tab2_combobox_2 =
59         ↪ QtWidgets.QComboBox(self.widget1)
59     self.tab2_combobox_2.setObjectName("tab2_combobox_2")
60     self.horizontalLayout_3.addWidget(self.tab2_combobox_
61         ↪ 2)
61     spacerItem3 = QtWidgets.QSpacerItem(90, 20,
62         ↪ QtWidgets.QSizePolicy.Maximum,
63         ↪ QtWidgets.QSizePolicy.Minimum)
62     self.horizontalLayout_3.addItem(spacerItem3)
63     self.tab2_label_2 = QtWidgets.QLabel(self.splitter_2)
64     self.tab2_label_2.setObjectName("tab2_label_2")
65     self.tabWidget.addTab(self.tab_2, "")
66     self.tab_3 = QtWidgets.QWidget()
67     self.tab_3.setObjectName("tab_3")
68     self.splitter_3 = QtWidgets.QSplitter(self.tab_3)
69     self.splitter_3.setGeometry(QtCore.QRect(0, 10, 361,
70         ↪ 221))
70     self.splitter_3.setOrientation(QtCore.Qt.Vertical)
```

```
71     self.splitter_3.setObjectName("splitter_3")
72     self.widget2 = QtWidgets.QWidget(self.splitter_3)
73     self.widget2.setObjectName("widget2")
74     self.horizontalLayout_4 =
75         ↳ QtWidgets.QHBoxLayout(self.widget2)
76     self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
77     self.horizontalLayout_4.setObjectName("horizontalLayo_
78         ↳ ut_4")
79     spacerItem4 = QtWidgets.QSpacerItem(90, 20,
80         ↳ QtWidgets.QSizePolicy.Maximum,
81         ↳ QtWidgets.QSizePolicy.Minimum)
82     self.horizontalLayout_4.addItem(spacerItem4)
83     self.tab3_combobox_3 =
84         ↳ QtWidgets.QComboBox(self.widget2)
85     self.tab3_combobox_3.setObjectName("tab3_combobox_3")
86     self.horizontalLayout_4.addWidget(self.tab3_combobox_
87         ↳ 3)
88     spacerItem5 = QtWidgets.QSpacerItem(90, 20,
89         ↳ QtWidgets.QSizePolicy.Maximum,
90         ↳ QtWidgets.QSizePolicy.Minimum)
91     self.horizontalLayout_4.addItem(spacerItem5)
92     self.tab3_label_3 = QtWidgets.QLabel(self.splitter_3)
93     self.tab3_label_3.setObjectName("tab3_label_3")
94     self.tabWidget.addTab(self.tab_3, "")
95     self.tab_4 = QtWidgets.QWidget()
96     self.tab_4.setObjectName("tab_4")
97     self.progressBar = QtWidgets.QProgressBar(self.tab_4)
98     self.progressBar.setGeometry(QtCore.QRect(90, 120,
99         ↳ 191, 21))
100     self.progressBar.setProperty("value", 24)
101     self.progressBar.setObjectName("progressBar")
102     self.widget3 = QtWidgets.QWidget(self.tab_4)
103     self.widget3.setGeometry(QtCore.QRect(0, 10, 361, 22))
104     self.widget3.setObjectName("widget3")
105     self.horizontalLayout_5 =
106         ↳ QtWidgets.QHBoxLayout(self.widget3)
107     self.horizontalLayout_5.setContentsMargins(0, 0, 0, 0)
108     self.horizontalLayout_5.setObjectName("horizontalLayo_
109         ↳ ut_5")
```

```

99         spacerItem6 = QtWidgets.QSpacerItem(90, 20,
    ↪     QtWidgets.QSizePolicy.Maximum,
    ↪     QtWidgets.QSizePolicy.Minimum)
100     self.horizontalLayout_5.addItem(spacerItem6)
101     self.tab3_combobox_4 =
    ↪     QtWidgets.QComboBox(self.widget3)
102     self.tab3_combobox_4.setObjectName("tab3_combobox_4")
103     self.horizontalLayout_5.addWidget(self.tab3_combobox_
    ↪     4)
104     spacerItem7 = QtWidgets.QSpacerItem(90, 20,
    ↪     QtWidgets.QSizePolicy.Maximum,
    ↪     QtWidgets.QSizePolicy.Minimum)
105     self.horizontalLayout_5.addItem(spacerItem7)
106     self.tabWidget.addTab(self.tab_4, "")
107     self.OK = QtWidgets.QPushButton(Form)
108     self.OK.setGeometry(QtCore.QRect(254, 270, 111, 23))
109     self.OK.setObjectName("OK")
110
111     self.retranslateUi(Form)
112     self.tabWidget.setCurrentIndex(0)
113     QtCore.QMetaObject.connectSlotsByName(Form)
114
115     def retranslateUi(self, Form):
116         _translate = QtCore.QCoreApplication.translate
117         Form.setWindowTitle(_translate("Form", "Form"))
118         self.tab1_label.setText(_translate("Form",
    ↪     "TextLabel"))
119         self.tabWidget.setTabText(self.tabWidget.indexOf(self
    ↪     .tab), _translate("Form", "Tab
    ↪     1"))
120         self.tab2_label_2.setText(_translate("Form",
    ↪     "TextLabel"))
121         self.tabWidget.setTabText(self.tabWidget.indexOf(self
    ↪     .tab_2), _translate("Form", "Tab
    ↪     2"))
122         self.tab3_label_3.setText(_translate("Form",
    ↪     "TextLabel"))

```

```

123         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪ .tab_3), _translate("Form", "Tab
        ↪ 3"))
124     self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪ .tab_4), _translate("Form", "Tab
        ↪ 4"))
125     self.OK.setText(_translate("Form", "PushButton"))
126

```

1.8 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.SetCurrentProcessExplicitAppUse
        ↪ rModelID("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_())

```

1.9 CvPyGui/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_path",
            ↳ 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", "record") # 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,
            ↳ 把之前的路径赋给新的路径
36            if m == "":
37                m = former_save_path
38            print(m)
39            savepath["save_path"] = m
40            # 将更改后 (或获取空时不更改) 的路径写入 json 文件

```

```
41         with open("config/db.json","w",encoding='UTF-8') as
           ↳ dbfile:
42             json.dump(savepath,dbfile)
43         time.sleep(0.5)
44         self.update_file_save_path()
45     def update_file_save_path(self):
46         with open("config/db.json","r",encoding='UTF-8') as
           ↳ dbfile_r:
47             file_save_path = json.load(dbfile_r)
48             self.file_path_label.setText(str(file_save_path['save_
           ↳ _path']))
49     def ok_close(self):
50         self.close()
51     def mp3_button(self):
52         if self.radioButton_mp3.isChecked() == True:
53             self.voiceformat = ".mp3"
54         else:
55             self.voiceformat = ".wav"
56         with open("config/db.json","r",encoding='UTF-8') as
           ↳ dbfile:
57             json_file = json.load(dbfile)
58             json_file["voice_format"] = self.voiceformat
59         with open("config/db.json","w",encoding='UTF-8') as
           ↳ dbfile:
60             json.dump(json_file,dbfile)
61         print(self.voiceformat)
62     def video_radio_button(self):
63         if self.radioButton_mp4.isChecked() == True:
64             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)

```

1.10 CvPyGui/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.layout.addWidget(self.name_lbl)
52         self.layout.addWidget(self.k_lbl[0])
53         self.layout.addWidget(self.thresh_sld)
54         self.layout.addWidget(self.delete_filter_btn)
55     def changeValue(self, value):
56         # Function for setting the value of k1
57         if value % 2 == 1:
58             self.k[0] = value
59         else:
60             self.k[0] = value + 1
61         self.thresh_sld.setValue(self.k[0])
62         self.k_lbl[0].setText(str(self.k[0]))
63         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,
        ↪ Y, k,
        ↪ 2)
84         elif name == 'HSV':
85             cv_before = cv2.cvtColor(cv_before,
        ↪ cv2.COLOR_RGB2HSV)
86             lower_color = np.array([k - 35, 0, 0])
87             upper_color = np.array([k + 35, 255, 255])
88             cv_after = cv2.inRange(cv_before, lower_color,
        ↪ upper_color)

```

```
89     elif name == 'LAB':
90         cv_before = cv2.cvtColor(cv_before,
91             ↪ cv2.COLOR_RGB2LAB)
92         L, a, b = cv2.split(cv_before)
93         ret, cv_after = cv2.threshold(L, k, 255,
94             ↪ cv2.THRESH_BINARY)
95     elif name == 'Erosion':
96         cv_before = cv2.cvtColor(cv_before,
97             ↪ cv2.COLOR_RGB2GRAY)
98         cv_after = cv2.erode(cv_before, kernel,
99             ↪ iterations=1)
100     elif name == 'Dilation':
101         cv_before = cv2.cvtColor(cv_before,
102             ↪ cv2.COLOR_RGB2GRAY)
103         cv_after = cv2.dilate(cv_before, kernel,
104             ↪ iterations=1)
105     elif name == 'Opening':
106         cv_before = cv2.cvtColor(cv_before,
107             ↪ cv2.COLOR_RGB2GRAY)
108         cv_after = cv2.morphologyEx(
109             ↪ cv_before, cv2.MORPH_OPEN, kernel)
110     elif name == 'Closing':
111         cv_before = cv2.cvtColor(cv_before,
112             ↪ cv2.COLOR_RGB2GRAY)
113         cv_after = cv2.morphologyEx(
114             ↪ cv_before, cv2.MORPH_CLOSE, kernel)
115     elif name == 'Top Hat':
116         cv_before = cv2.cvtColor(cv_before,
117             ↪ cv2.COLOR_RGB2GRAY)
118         cv_after = cv2.morphologyEx(
119             ↪ cv_before, cv2.MORPH_TOPHAT, kernel)
120     elif name == 'Black Hat':
121         cv_before = cv2.cvtColor(cv_before,
122             ↪ cv2.COLOR_RGB2GRAY)
123         cv_after = cv2.morphologyEx(
124             ↪ cv_before, cv2.MORPH_BLACKHAT, kernel)
125     elif name == 'Canny':
126         cv_before = cv2.cvtColor(cv_before,
127             ↪ cv2.COLOR_RGB2GRAY)
```

```

117         cv_after = cv2.Canny(cv_before, 100, k)
118     elif name == 'Laplacian':
119         cv_before = cv2.cvtColor(cv_before,
120             ↪ cv2.COLOR_RGB2GRAY)
121         cv_after = cv2.Laplacian(cv_before, cv2.CV_64F)
122         cv_after = np.absolute(cv_after)
123         cv_after = np.uint8(cv_after)
124     return cv_after

```

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

```

1.12 CvPyGui/___init___py

1

1.13 CvPyGui/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_
                    ↪ 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.addItem(cameralist_child)
99         self.tab1_combobox_4.addItem(cameralist_child)
100        self.CreateButtons()
101        with open("config/db.json","r",encoding='UTF-8') as
102            ↪ dbfile_r:
103            camera_voice_name = json.load(dbfile_r)
104            self.tab1_text = camera_voice_name["hd_camera_name"]
105            self.tab2_text = camera_voice_name["face_camera_name"]
106            self.tab3_text = camera_voice_name["eye_camera_name"]
107            self.tab4_text = camera_voice_name["hk_voice_name"]
108            if len(cameralist_child) >= 1:
109                for i in range(len(cameralist_child)):
110                    camera_judge[i] = 'close'
111            else:
112                print('no camera')
113        p = pyaudio.PyAudio()
114        info = p.get_host_api_info_by_index(0)
115        numdevices = info.get('deviceCount')
116        global voice_list
117        voice_list = []
118        for i in range(0, numdevices):
119            if (p.get_device_info_by_host_api_device_index(0,
120                ↪ i).get('maxInputChannels')) > 0:
121                voice_list.append(p.get_device_info_by_host_a
122                ↪ pi_device_index(0,
123                ↪ i).get('name'))
124        self.tab1_combobox_7.addItem(voice_list)
125        self.child_wind_judge = True
126        if self.tab1_text in cameralist_child:
127            self.tab1_combobox.setCurrentIndex(cameralist_chi
128            ↪ ld.index(self.tab1_text) +
129            ↪ 2)
130        if self.tab2_text in cameralist_child:
131            self.tab1_combobox_3.setCurrentIndex(cameralist_c
132            ↪ hild.index(self.tab2_text) +
133            ↪ 2)
134        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,
        ↪ (tab1_camera_index,))
148     else:
149         self.update_tab1_image.updateImage(img_no_camera)
150     def tab2_combobox_setting(self):
151         tab2_camera_text = self.tab1_combobox_3.currentText()
152         self.tab2_text = tab2_camera_text
153     try:
154         tab2_camera_index =
        ↪ cameralist_child.index(tab2_camera_text)
155         tab2_judge = True

```



```

156         except:
157             self.update_tab2_image.updateImage(img_no_camera)
158             tab2_judge = False
159             # 获取摄像头的名字后显示画面
160             # 启动一个线程，显示摄像头捕获的画面
161             if tab2_judge:
162                 # 如果选择的摄像头没有打开
163                 if camera_judge[tab2_camera_index] == 'close':
164                     camera_judge[tab2_camera_index] = 'open'
165                     threading._start_new_thread(self.open_camera_,
166                                                 ↪ 2,
167                                                 ↪ (tab2_camera_index,))
168             else:
169                 self.update_tab2_image.updateImage(img_no_camera)
170
171     def open_camera(self, camera_index):
172         #传入摄像头的序号，
173         ↪ 判断摄像头是否开启的代码应该在调用该函数的函数中。
174         #根据摄像头的序号，打开摄像头，
175         ↪ 然后把摄像头的画面传入其他函数？
176         #只有三个要显示的标签，判断要在那个标签上显示，
177         ↪ 然后更新图片。
178         # cap[camera_index] =
179         ↪ cv2.VideoCapture(camera_index, cv2.CAP_DSHOW)
180         cap[camera_index] = cv2.VideoCapture(camera_index)
181         print(cap[camera_index].get(3))
182         camera_judge[camera_index] = 'open'
183         cap[camera_index].set(3,hd_width)
184         cap[camera_index].set(4,hd_width)
185         cap[camera_index].set(5,30)
186         while self.child_wind_judge:
187             self.ret, frame[camera_index] =
188             ↪ cap[camera_index].read()
189         if self.ret:
190             #判断需要在哪几个标签上显示 判断方式：
191             ↪ 获取三个下拉列表当前显示的值，
192             ↪ 确定该值在摄像头列表中所占的序号，
193             ↪ 序号与 camera_index 相同则更新图片。

```

```

183         frame[camera_index] =
            ↪ cv2.cvtColor(frame[camera_index],
            ↪ cv2.COLOR_BGR2RGB)
184         if (self.tab1_combobox.currentText() !=
            ↪ place_choice_camera) and
            ↪ (self.tab1_combobox.currentText() !=
            ↪ no_camera_):
185             if camera_index == cameralist_child.index_1
            ↪ (self.tab1_combobox.currentText()):
186                 self.update_tab1_image.updateImage(fr_
            ↪ ame[camera_index])
187             else:
188                 pass
189         else:
190             pass
191         if (self.tab1_combobox_3.currentText() !=
            ↪ place_choice_camera) and
            ↪ (self.tab1_combobox_3.currentText() !=
            ↪ no_camera_):
192             if camera_index == cameralist_child.index_1
            ↪ (self.tab1_combobox_3.currentText()):
193                 self.update_tab2_image.updateImage(fr_
            ↪ ame[camera_index])
194             else:
195                 pass
196         else:
197             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_1
            ↪ (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() !=
                    ↪ pplace_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] =
251     ↪ cv2.VideoCapture(camera_index,cv2.CAP_DSHOW)
252     cap3[camera_index] = cv2.VideoCapture(camera_index)
253     camera_judge[camera_index] = 'open'
254     cap3[camera_index].set(3,hd_width)
255     cap3[camera_index].set(4,hd_width)
256     cap3[camera_index].set(5,30.0)
257     while self.child_wind_judge:
258         self.ret_3, frame3[camera_index] =
259         ↪ cap3[camera_index].read()
260         if self.ret_3:
261             frame3[camera_index] =
262             ↪ cv2.cvtColor(frame3[camera_index],
263             ↪ cv2.COLOR_BGR2RGB)
264             if (self.tab1_combobox.currentText() !=
265             ↪ pplace_choice_camera) and
266             ↪ (self.tab1_combobox.currentText() !=
267             ↪ no_camera_):
268                 if camera_index == cameralist_child.index_1
269                 ↪ (self.tab1_combobox.currentText()):
270                     self.update_tab1_image.updateImage(fr_
271                     ↪ ame3[camera_index])
272             else:
273                 pass
274         else:
275             pass
276         if (self.tab1_combobox_3.currentText() !=
277         ↪ pplace_choice_camera) and
278         ↪ (self.tab1_combobox_3.currentText() !=
279         ↪ no_camera_):
280             if camera_index == cameralist_child.index_2
281             ↪ (self.tab1_combobox_3.currentText()):
282                 self.update_tab2_image.updateImage(fr_
283                 ↪ ame3[camera_index])
284             else:
285                 pass
286         else:
287             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、
    ↪ 眼部的摄像头名字，初始内容为读取json文件得到的。
332     with open("config/db.json","r",encoding='UTF-8') as
    ↪ dbfile_r:
333         camera_voice_name = json.load(dbfile_r)
334         camera_voice_name["hd_camera_name"] = self.tab1_text
335         camera_voice_name["face_camera_name"] = self.tab2_text

```

```

336         camera_voice_name["eye_camera_name"] = self.tab3_text
337         camera_voice_name["hk_voice_name"] = self.tab4_text
338         with open("config/db.json", "w", encoding='UTF-8') as
            ↳ dbfile:
339             json.dump(camera_voice_name, dbfile)
340         self.child_wind_judge = False
341         try:
342             self.sub_thread.vo_judge = False
343             print('success')
344         except:
345             print('default')
346         self.close()# 关闭该窗口
347     def closeEvent(self, event):
348         print('关闭窗口')
349         self.child_wind_judge = False
350         try:
351             self.sub_thread.vo_judge = False
352             self.child_wind_judge = False
353         except:
354             pass
355     #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?
    ↳ 点确认时关闭, 先把选好的信息存起来, 点确认时修改。
356     def CreateButtons(self):
357         # print("createbuttons")
358         self.tab1_combobox.currentIndexChanged.connect(self.t
            ↳ 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)

```

1.14 CvPyGui/main.py

```

1  import wave
2  import cv2, os

```



```

3  from PyQt5.QtCore import Qt, QTimer
4  from PyQt5.QtWidgets import (QMainWindow, QMessageBox,)
5  import threading
6  import time
7  from CvPyGui import ImageCvQtContainer
8  from CvPyGui import child_CameraVoice_setting
9  from CvPyGui import Child_File_Setting
10 from CvPyGui.ui import gui
11 from datetime import datetime
12 import pygame
13 import pygame.camera
14 import pyaudio
15
16 import imutils
17 from imutils import face_utils
18 import dlib
19 import time
20 import numpy as np
21 from keras.models import load_model
22 from keras.preprocessing.image import img_to_array
23
24 from PIL import Image, ImageDraw, ImageFont
25
26 from mediapipe import solutions
27 from mediapipe.framework.formats import landmark_pb2
28
29 # STEP 1: Import the necessary modules.
30 import mediapipe as mp
31 from mediapipe.tasks import python
32 from mediapipe.tasks.python import vision
33
34 # STEP 2: Create an PoseLandmarker object.
35 base_options = python.BaseOptions(model_asset_path='pose_landmark_
    ↪ marker.task')
36 options = vision.PoseLandmarkerOptions(
37     base_options=base_options,
38     output_segmentation_masks=True)
39 detector = vision.PoseLandmarker.create_from_options(options)
40

```

```

41 Ui_MainWindow = gui.Ui_MainWindow
42 import json
43 img_no_camera = cv2.imread('config/image/no_camera.jpg',
    ↪ cv2.IMREAD_COLOR)
44 # img_no_voice = cv2.imread('config/image/no_voice.jpg',
    ↪ cv2.IMREAD_COLOR)
45 img_no_voice = cv2.imread('config/image/no_camera.jpg',
    ↪ cv2.IMREAD_COLOR)
46 img_voice=cv2.imread('config/image/voice.jpg',cv2.IMREAD_COLOR)
    ↪ R)
47
48
49 def preprocess_face(face):
50     face = cv2.cvtColor(face, cv2.COLOR_BGR2GRAY) # Convert
    ↪ to grayscale
51     face = cv2.resize(face, (64, 64)) # Resize to the
    ↪ required size
52     face = face.astype("float") / 255.0 # Normalize pixel
    ↪ values
53     face = img_to_array(face)
54     face = np.expand_dims(face, axis=0)
55     return face
56
57 def cv2PutText(img, text, postion, textColor=(0, 255, 0),
    ↪ textSize=30):
58     if (isinstance(img,np.ndarray)):
59         img = Image.fromarray(cv2.cvtColor(img,
    ↪ cv2.COLOR_BGR2RGB))
60
61     draw = ImageDraw.Draw(img)
62     font = ImageFont.truetype("font/simsun.ttc", textSize,
    ↪ encoding="utf-8")
63     draw.text(postion, text, textColor, font=font)
64     img = cv2.cvtColor(np.asarray(img), cv2.COLOR_RGB2BGR)
65     return img
66
67 def draw_landmarks_on_image(rgb_image, detection_result):
68     pose_landmarks_list = detection_result.pose_landmarks
69     annotated_image = np.copy(rgb_image)

```

```

70
71     # Loop through the detected poses to visualize.
72     for idx in range(len(pose_landmarks_list)):
73         pose_landmarks = pose_landmarks_list[idx]
74
75     # Draw the pose landmarks.
76     pose_landmarks_proto =
77         ↪ landmark_pb2.NormalizedLandmarkList()
78     pose_landmarks_proto.landmark.extend([
79         ↪ landmark_pb2.NormalizedLandmark(x=landmark.x,
80         ↪ y=landmark.y, z=landmark.z) for landmark in
81         ↪ pose_landmarks
82     ])
83     solutions.drawing_utils.draw_landmarks(
84         annotated_image,
85         pose_landmarks_proto,
86         solutions.pose.POSE_CONNECTIONS,
87         solutions.drawing_styles.get_default_pose_landmarks_style(),
88         ↪ le())
89
90     return annotated_image
91
92
93 class MyApp(QMainWindow, Ui_MainWindow, threading.Thread):
94     filter_count = 0
95     def __init__(self, chunk=1024, rate=16000):
96         super().__init__()
97         threading.Thread.__init__(self)
98         QMainWindow.__init__(self)
99         Ui_MainWindow.__init__(self)
100        self.setupUi(self)
101        self.initUI()
102        self.hd_camera=-1
103        self.voice_index=-1
104        self.eye_camera=-1
105        self.face_camera=-1
106        self.hd_width = 1920
107        self.hd_height = 1080
108        self.face_width = 1280
109        self.eye_width = 1280
110        self.face_height = 1920

```

```

105         self.eye_height = 720
106         self.hd_fps = 30.0
107         self.face_fps = 10.0
108         self.eye_fps = 10.0
109         self.CHUNK = chunk
110         self.FORMAT = pyaudio.paInt16
111         self.CHANNELS = 1
112         self.RATE = rate
113         self._win1_running = True
114         self._win2_running = True
115         self._win3_running = True
116         self._voice_get = True
117         self._begin = False
118         self._frames = []
119         self._init_running = True
120         self._start_running = True
121         self.record_judge = False
122         self.recording_time = '0'
123         self.recording_location = 'record/'
124         self.recording_information = self.recording_time + '
    ↪ ' + self.recording_location
125     def initUI(self):
126         self.original1_image = ImageCvQtContainer.Image(
127             'camera1', self.original_frame_lbl)
128         self.original2_image = ImageCvQtContainer.Image(
129             'camera2', self.processed_frame_lbl)
130         self.eye_image =
131             ↪ ImageCvQtContainer.Image('eye_camera',
132             ↪ self.eye_lbl)
131         self.voice_image =
132             ↪ ImageCvQtContainer.Image('voice_mic',
133             ↪ self.voice_lbl)
132         self.setBackground()
133         self.createButtons()
134         self.Child_File_Setting()
135     def r_json(self):
136         # 读取配置文件
137         with open('config/db.json', encoding='utf-8', mode='r')
138             ↪ as rf:

```

```

138         data = rf.read().encode()
139         configuration = json.loads(data)
140         print('configuration')
141         print(configuration)
142         self.hd_cream_name=configuration['hd_camera_name']
143         self.face_camera_name=configuration['face_camera_
↳ name']
144         self.eye_camera_name=configuration['eye_camera_na
↳ me']
145         self.vocie_name=configuration['voice_name']
146         self.hk_voice_name=configuration['hk_voice_name']
147         self.bluetooth_voice_name=configuration['bluetoot
↳ h_voice_name']
148         self.save_path=configuration['save_path']
149         self.hd_width=configuration['hd_width']
150         self.hd_height=configuration['hd_height']
151         self.hd_fps=configuration['hd_fps']
152         self.face_width=configuration['face_width']
153         self.face_height=configuration['face_height']
154         self.face_fps=configuration['face_fps']
155         self.eye_width=configuration['eye_width']
156         self.eye_height=configuration['eye_height']
157         self.eye_fps=configuration['eye_fps']
158         self.voice_format = configuration['voice_format']
159         self.video_format = configuration['video_format']
160         self.recorfing_location = self.save_path
161     def initfrom(self):
162         if self._init_running:
163             self._init_running=False
164             pygame.init()
165             pygame.camera.init()
166             cameralist = pygame.camera.list_cameras()
167             print(cameralist)
168             p = pyaudio.PyAudio()
169             info = p.get_host_api_info_by_index(0)
170             numdevices = info.get('deviceCount')
171             voice_list = []
172             for i in range(0, numdevices):

```

```

173         if (p.get_device_info_by_host_api_device_index(
174             ↪ x(0, i).get('maxInputChannels')) >
175             ↪ 0:
176             voice_list.append(p.get_device_info_by_host_api_device_index(0,
177                 ↪ i).get('name'))
178     print(voice_list)
179     try:
180         self.r_json()
181     except:
182         self.statusbar.showMessage('未找到配置文件', 5000)
183     try:
184         self.hd_camera=cameralist.index(self.hd_camera_name)
185     except:
186         self.hd_camera=-1
187     try:
188         self.face_camera=cameralist.index(self.face_camera_name)
189     except:
190         self.face_camera=-1
191     try:
192         self.eye_camera=cameralist.index(self.eye_camera_name)
193     except:
194         self.eye_camera=-1
195     try:
196         if self.hk_voice_name in voice_list:
197             self.voice_index =
198                 ↪ voice_list.index(self.hk_voice_name)
199         else:
200             if self.vocie_name in voice_list and
201                 ↪ self.bluetooth_voice_name in
202                 ↪ voice_list:
203                 self.voice_index=voice_list.index(self.vocie_name)
204             self.CHANNELS=2

```

```

199         elif self.vocie_name in voice_list and
        ↪ self.bluetooth_voice_name not in
        ↪ voice_list:
200             self.voice_index=voice_list.index(self
        ↪ f.vocie_name)
201             self.CHANNELS=2
202         elif self.vocie_name not in voice_list and
        ↪ self.bluetooth_voice_name in
        ↪ voice_list:
203             self.voice_index=voice_list.index(self
        ↪ f.bluetooth_voice_name)
204         else:
205             self.voice_index = -1
206     except:
207         self.voice_index=-1
208     print(self.hd_camera,'行为设备 index')
209     print(self.face_camera,'面部设备 index')
210     print(self.eye_camera,'眼部设备 index')
211     print(self.voice_index,'音频设备 index')
212     if os.path.exists(self.save_path) is False:
213         os.makedirs(self.save_path)
214     if self.hd_camera == -1 and self.face_camera == -1
        ↪ and self.eye_camera ==-1 and self.voice_index
        ↪ ==-1:
215         self.statusbar.showMessage('未检测到任何可用设
        ↪ 备',
        ↪ 5000)
216     elif self.hd_camera != -1 and self.face_camera ==
        ↪ -1 and self.eye_camera ==-1 and
        ↪ self.voice_index ==-1:
217         self.statusbar.showMessage('未检测到面部、眼部、
        ↪ 音频设备',5000)
218     elif self.hd_camera == -1 and self.face_camera !=
        ↪ -1 and self.eye_camera ==-1 and
        ↪ self.voice_index ==-1:
219         self.statusbar.showMessage('未检测到行为、眼部、
        ↪ 音频设备', 5000)

```

```
220         elif self.hd_camera == -1 and self.face_camera ==
↳ -1 and self.eye_camera !=-1 and
↳ self.voice_index ==-1:
221             self.statusbar.showMessage('未检测到行为、面部、
↳ 音频设备', 5000)
222         elif self.hd_camera == -1 and self.face_camera ==
↳ -1 and self.eye_camera ==-1 and
↳ self.voice_index !=-1:
223             self.statusbar.showMessage('未检测到行为、面部、
↳ 眼部设备', 5000)
224         elif self.hd_camera != -1 and self.face_camera !=
↳ -1 and self.eye_camera ==-1 and
↳ self.voice_index ==-1:
225             self.statusbar.showMessage('未检测到眼部、
↳ 音频设备', 5000)
226         elif self.hd_camera != -1 and self.face_camera ==
↳ -1 and self.eye_camera !=-1 and
↳ self.voice_index ==-1:
227             self.statusbar.showMessage('未检测到面部、
↳ 音频设备', 5000)
228         elif self.hd_camera != -1 and self.face_camera ==
↳ -1 and self.eye_camera ==-1 and
↳ self.voice_index !=-1:
229             self.statusbar.showMessage('未检测到面部、
↳ 眼部设备', 5000)
230         elif self.hd_camera == -1 and self.face_camera !=
↳ -1 and self.eye_camera !=-1 and
↳ self.voice_index ==-1:
231             self.statusbar.showMessage('未检测到行为、
↳ 音频设备', 5000)
232         elif self.hd_camera == -1 and self.face_camera !=
↳ -1 and self.eye_camera ==-1 and
↳ self.voice_index !=-1:
233             self.statusbar.showMessage('未检测到行为、
↳ 眼部设备', 5000)
234         elif self.hd_camera == -1 and self.face_camera ==
↳ -1 and self.eye_camera !=-1 and
↳ self.voice_index !=-1:
```



```

235         self.statusbar.showMessage('未检测到行为、
        ↪ 面部设备', 5000)
236     elif self.hd_camera != -1 and self.face_camera !=
        ↪ -1 and self.eye_camera == -1 and
        ↪ self.voice_index != -1:
237         self.statusbar.showMessage('未检测到眼部设备',
        ↪ 5000)
238     elif self.hd_camera != -1 and self.face_camera ==
        ↪ -1 and self.eye_camera != -1 and
        ↪ self.voice_index != -1:
239         self.statusbar.showMessage('未检测到面部设备',
        ↪ 5000)
240     elif self.hd_camera == -1 and self.face_camera !=
        ↪ -1 and self.eye_camera != -1 and
        ↪ self.voice_index != -1:
241         self.statusbar.showMessage('未检测到行为设备',
        ↪ 5000)
242     elif self.hd_camera != -1 and self.face_camera !=
        ↪ -1 and self.eye_camera != -1 and
        ↪ self.voice_index == -1:
243         self.statusbar.showMessage('未检测到音频设备',
        ↪ 5000)
244     else:
245         self.statusbar.showMessage('所有设备准备就绪',
        ↪ 5000)
246         self.start()
247         # self.win_4_upimage()
248     else:
249         pass
250 def child_CameraVoice_setting(self):#点击此选项时不能处于「
    ↪ 以下状态： 录制视频中。
    ↪
251     if self._init_running == False:
252         QMessageBox.information(self, '警告', '进程未中止,
        ↪ 请中止进程再点击此选项! ')
253     else:
254         # 结束录制
255         self.endRe()
256         self.f_p()

```

```

257         self.Child_window =
            ↳ child_CameraVoice_setting.Child_Window()
258         self.Child_window.setWindowModality(Qt.Applicatio
            ↳ nModal)
259         self.Child_window.show()
260     def Child_File_Setting(self):
261         # 因为打开软件时便会调用该函数，所以不能调用下面两个方法
262         self.child_file_window =
            ↳ Child_File_Setting.child_file_setting()
263         self.child_file_window.setWindowModality(Qt.Applicati
            ↳ onModal)
264         self.child_file_window.show()
265     def start(self):
266         threading._start_new_thread(self.win_1,())
267         # threading._start_new_thread(self.win_2,())
268         # threading._start_new_thread(self.win_3,())
269     def win_1(self):
270         if self.hd_camera != -1:
271             self._win1_running=True
272             self.cap1 = 0
273             # self.cap1 =
            ↳ cv2.VideoCapture(self.hd_camera,cv2.CAP_DSHOW)
274             # self.cap1 = cv2.VideoCapture(self.hd_camera)
275             self.cap1 = cv2.VideoCapture("elder_dance.avi")
276             # self.cap1.set(3,2560)
277             # self.cap1.set(4,1440)
278             # self.cap1.set(5,self.hd_fps)
279             print(self.cap1.get(3))
280             print(self.cap1.get(4))
281             self.cap1_3 = self.cap1.get(3) # 宽
282             self.cap1_4 = self.cap1.get(4) # 高
283             self.hd_width = self.cap1_3
284             self.hd_height = self.cap1_4
285             with open("config/db.json", "r", encoding='UTF-8')
            ↳ as dbfile_r_hd:
286                 jsonfile_hd = json.load(dbfile_r_hd)
287                 jsonfile_hd['hd_width'] = self.hd_width
288                 jsonfile_hd['hd_height'] = self.hd_height
289                 if self.face_camera_name == self.hd_cream_name:

```

```

290         self.face_width = self.cap1_3
291         self.face_height = self.cap1_4
292         jsonfile_hd['face_width'] = self.face_width
293         jsonfile_hd['face_height'] = self.face_height
294     if self.hd_cream_name == self.eye_camera_name:
295         self.eye_width = self.cap1_3
296         self.eye_height = self.cap1_4
297         jsonfile_hd['eye_width'] = self.eye_width
298         jsonfile_hd['eye_height'] = self.eye_height
299         print('toooooo')
300     with open("config/db.json", "w", encoding='UTF-8')
    ↪ as dbfile_hd:
301         json.dump(jsonfile_hd, dbfile_hd)
302     self.cap1.set(3, self.hd_width)
303     self.cap1.set(4, self.hd_width)
304
305     while self._win1_running:
306         ret, frame = self.cap1.read()
307         if ret:
308             frame = cv2.cvtColor(frame,
    ↪ cv2.COLOR_BGR2RGB)
309             self.original1_image.updateImage(frame,
    ↪ e)
310
311             mp_image = mp.Image(image_format=mp.I_
    ↪ mageFormat.SRGB,
    ↪ data=frame)
312
313             # STEP 4: Detect pose landmarks from
    ↪ the input image.
314             detection_result =
    ↪ detector.detect(mp_image)
315
316             blank_image = frame.copy()
317             blank_image.fill(0)
318             # STEP 5: Process the detection
    ↪ result. In this case, visualize
    ↪ it.

```

```

319         annotated_image =
320             ↪ draw_landmarks_on_image(frame,
321             ↪ detection_result)
322         blank_image = draw_landmarks_on_image(
323             ↪ (blank_image,
324             ↪ detection_result)
325         self.eye_image.updateImage(annotated_
326             ↪ image)
327         self.original2_image.updateImage(blan
328             ↪ k_image)
329
330         visualized_mask = None
331         if detection_result.segmentation_masks
332             ↪ is not None:
333             rgb = cv2.cvtColor(detection_resu
334             ↪ lt.segmentation_masks[0].numpy
335             ↪ y_view(),
336             ↪ cv2.COLOR_GRAY2RGB)
337             rgb = rgb * 255
338             rgb = rgb.astype(np.uint8)
339             visualized_mask = rgb
340         else:
341             black = np.zeros(frame.shape,
342             ↪ dtype=np.uint8)
343             visualized_mask = black
344
345         self.voice_image.updateImage(visualiz
346             ↪ ed_mask)
347
348         # ff.write(str(id) + "\n" +
349             ↪ np.array2string(shape) + "\n")
350
351         # if self.face_camera_name == self.hd_cream_name
352             ↪ and self.hd_cream_name ==
353             ↪ self.eye_camera_name:
354             # while self._win1_running:
355             #     ret, frame = self.cap1.read()
356             #     if ret:

```

```

343         # frame = cv2.cvtColor(frame,
↪ cv2.COLOR_BGR2RGB)
344     #
↪ self.original1_image.updateImage(frame)
345     # # self.eye_image.updateImage(frame)
346     #
↪ self.original2_image.updateImage(frame)
347     # if self.hd_cream_name == self.face_camera_name
↪ and self.hd_cream_name !=
↪ self.eye_camera_name:
348     # while self._win1_running:
349     #     ret, frame = self.cap1.read()
350     #     if ret:
351     #         frame = cv2.cvtColor(frame,
↪ cv2.COLOR_BGR2RGB)
352     #
↪ self.original1_image.updateImage(frame)
353     #
↪ self.original2_image.updateImage(frame)
354     # if self.hd_cream_name != self.face_camera_name
↪ and self.hd_cream_name ==
↪ self.eye_camera_name:
355     # while self._win1_running:
356     #     ret, frame = self.cap1.read()
357     #     if ret:
358     #         frame = cv2.cvtColor(frame,
↪ cv2.COLOR_BGR2RGB)
359     #
↪ self.original1_image.updateImage(frame)
360     # self.eye_image.updateImage(frame)
361     # if self.hd_cream_name != self.face_camera_name
↪ and self.hd_cream_name !=
↪ self.eye_camera_name:
362     # while self._win1_running:
363     #     ret, frame = self.cap1.read()
364     #     if ret:
365     #         frame = cv2.cvtColor(frame,
↪ cv2.COLOR_BGR2RGB)

```

```

366         #
367         ↪ self.original1_image.updateImage(frame)
368
369         # self.original1_image.updateImage(img_no_camera)
370         # if self.face_camera_name == self.hd_cream_name:
371         #     self.original2_image.updateImage(img_no_camera)
372         ↪ era)
373         # if self.hd_cream_name == self.eye_camera_name:
374         #     self.eye_image.updateImage(img_no_camera)
375     else:
376         self.original1_image.updateImage(img_no_camera)
377 def win_2(self):
378     if (self.face_camera != -1) and (self.face_camera_name
379     ↪ != self.hd_cream_name):#脸部摄像头存在并且不和行为
380     ↪ 摄像头重合
381     self._win2_running=True
382     self.cap2 = 0
383     self.cap2 = cv2.VideoCapture(self.face_camera,cv2_
384     ↪ .CAP_DSHOW)
385     self.cap2.set(3,2560)
386     self.cap2.set(4,1440)
387     self.cap2.set(5,self.face_fps)
388     print('cap2'+ '_' +str(self.cap2.get(3)))
389     print('cap2'+ '_' +str(self.cap2.get(4)))
390     print('cap2'+ '_' +str(self.cap2.get(5)))
391     self.cap2_3 = self.cap2.get(3) # 宽
392     self.cap2_4 = self.cap2.get(4) # 高
393     self.face_width = self.cap2_3
394     self.face_height = self.cap2_4
395     with open("config/db.json", "r", encoding='UTF-8')
396     ↪ as dbfile_r_face:
397         jsonfile_face = json.load(dbfile_r_face)
398         jsonfile_face['face_width'] = self.face_width
399         jsonfile_face['face_height'] = self.face_height
400     if self.face_camera_name == self.eye_camera_name:
401         self.eye_width = self.cap2_3
402         self.eye_height = self.cap2_4
403         jsonfile_face['eye_width'] = self.eye_width
404         jsonfile_face['eye_height'] = self.eye_height

```

```

399         with open("config/db.json", "w", encoding='UTF-8')
        ↪ as dbfile_face:
400             json.dump(jsonfile_face, dbfile_face)
401         self.cap2.set(3, self.face_width)
402         self.cap2.set(4, self.face_width)
403         if self.face_camera_name == self.eye_camera_name:
404             while self._win2_running:
405                 ret2, frame2 = self.cap2.read()
406                 if ret2:
407                     frame2 = cv2.cvtColor(frame2,
        ↪ cv2.COLOR_BGR2RGB)
408                     self.original2_image.updateImage(frame2)
        ↪ e2)
409                     self.eye_image.updateImage(frame2)
410             else:
411                 while self._win2_running:
412                     ret2, frame2 = self.cap2.read()
413                     if ret2:
414                         frame2 = cv2.cvtColor(frame2,
        ↪ cv2.COLOR_BGR2RGB)
415                         self.original2_image.updateImage(frame2)
        ↪ e2)
416                 self.original2_image.updateImage(img_no_camera)
417                 if self.face_camera_name == self.eye_camera_name:
418                     self.eye_image.updateImage(img_no_camera)
419             else:
420                 self.original2_image.updateImage(img_no_camera)
421     def win_3(self):
422         if (self.eye_camera != -1) and (self.eye_camera_name
        ↪ != self.face_camera_name) and
        ↪ (self.eye_camera_name !=
        ↪ self.hd_cream_name): #不和脸部、行为重合
423             self._win3_running=True
424             self.cap3 = 0
425             self.cap3 = cv2.VideoCapture(self.eye_camera, cv2.
        ↪ CAP_DSHOW)
426             self.cap3.set(3, 2560)
427             self.cap3.set(4, 1440)
428             self.cap3.set(5, self.eye_fps)

```

```

429     print('cap3'+ '_' +str(self.cap3.get(3)))
430     print('cap3'+ '_' +str(self.cap3.get(4)))
431     print('cap3'+ '_' +str(self.cap3.get(5)))
432     self.cap3_3 = self.cap3.get(3)  # 宽
433     self.cap3_4 = self.cap3.get(4)  # 高
434     self.eye_width = self.cap3_3
435     self.eye_height = self.cap3_4
436     with open("config/db.json", "r", encoding='UTF-8')
        ↪ as dbfile_r_eye:
437         jsonfile_eye = json.load(dbfile_r_eye)
438         jsonfile_eye['face_width'] = self.eye_width
439         jsonfile_eye['face_height'] = self.eye_height
440     with open("config/db.json", "w", encoding='UTF-8')
        ↪ as dbfile_eye:
441         json.dump(jsonfile_eye, dbfile_eye)
442     self.cap3.set(3, self.eye_width)
443     self.cap3.set(4, self.eye_height)
444     while self._win3_running:
445         ret3, frame3 = self.cap3.read()
446         if ret3:
447             frame3 = cv2.cvtColor(frame3,
        ↪ cv2.COLOR_BGR2RGB)
448             self.eye_image.updateImage(frame3)
449             self.eye_image.updateImage(img_no_camera)
450         else:
451             self.eye_image.updateImage(img_no_camera)
452     def save_hd_video(self):
453         if self.hd_camera != -1:
454             fn = self.lineEdit.text()
455             name=self.save_path+"/behavior_" + fn + "_" +
        ↪ datetime.now().strftime('%Y%m%d%H%M%S') +
        ↪ self.video_format
456             fourcc = cv2.VideoWriter_fourcc(*'XVID')
457             out = cv2.VideoWriter(name, fourcc,
        ↪ int(self.hd_fps), (int(self.hd_width),
        ↪ int(self.hd_height)))
458             while self._begin:
459                 ret,frame=self.cap1.read()
460                 out.write(frame)

```



```

461     def save_face_video(self):
462         if self.face_camera != -1:
463             fn = self.lineEdit.text()
464             name=self.save_path+"/face_" + fn + "_" +
↳             datetime.now().strftime('%Y%m%d%H%M%S') +
↳             self.video_format
465             fourcc = cv2.VideoWriter_fourcc(*'XVID')
466             out = cv2.VideoWriter(name, fourcc,
↳             int(self.face_fps), (int(self.face_width),
↳             int(self.face_height)))
467             if self.face_camera_name != self.hd_cream_name:
468                 while self._begin:
469                     ret,frame=self.cap2.read()
470                     out.write(frame)
471             else:
472                 while self._begin:
473                     ret,frame=self.cap1.read()
474                     out.write(frame)
475     def save_eye_video(self):
476         if self.eye_camera != -1:
477             fn = self.lineEdit.text()
478             name=self.save_path+"/eye_" + fn + "_" +
↳             datetime.now().strftime('%Y%m%d%H%M%S') +
↳             self.video_format
479             fourcc = cv2.VideoWriter_fourcc(*'XVID')
480             out = cv2.VideoWriter(name, fourcc,
↳             int(self.eye_fps), (int(self.eye_width),
↳             int(self.eye_height)))
481             if (self.eye_camera_name != self.hd_cream_name) &
↳             (self.eye_camera_name !=
↳             self.face_camera_name):
482                 while self._begin:
483                     ret,frame=self.cap3.read()
484                     out.write(frame)
485             elif (self.eye_camera_name != self.hd_cream_name)
↳             & (self.eye_camera_name ==
↳             self.face_camera_name):
486                 while self._begin:
487                     ret,frame=self.cap2.read()

```

```

488             out.write(frame)
489         else:
490             while self._begin:
491                 ret, frame=self.cap1.read()
492                 out.write(frame)
493     def win_4_upimage(self):
494         if self.voice_index != -1:
495             self.voice_image.updateImage(img_voice)
496         else:
497             pass
498     def win_4(self):
499         self._voice_get = True
500         self._frames=[]
501         p=pyaudio.PyAudio()
502         print(self.voice_index)
503         stream=p.open(format=self.FORMAT,
504                        rate=self.RATE,
505                        channels=self.CHANNELS,
506                        input=True,
507                        frames_per_buffer=self.CHUNK,
508                        input_device_index=self.voice_index)
509         while self._voice_get:
510             data = stream.read(self.CHUNK)
511             self._frames.append(data)
512         stream.stop_stream()
513         stream.close()
514         p.terminate()
515     def voice_save(self):
516         p = pyaudio.PyAudio()
517         fn = self.lineEdit.text()
518         name = self.save_path+"/vioce_" + fn + "_" +
519             ↪ datetime.now().strftime('%Y%m%d%H%M%S') +
520             ↪ self.voice_format
521         wf = wave.open(name, 'wb')
522         wf.setnchannels(self.CHANNELS)
523         wf.setsampwidth(p.get_sample_size(self.FORMAT))
524         wf.setframerate(self.RATE)
525         wf.writeframes(b''.join(self._frames))
526         wf.close()

```

```

525         print("Saved")
526     def updatemessage(self):
527         # 只有在点击开始录制时才会显示该消息,
528         ↪ 点击进程结束时停止显示。
529         with open("config/db.json", "r", encoding='UTF-8') as
530             dbfile_r:
531                 jsonfile = json.load(dbfile_r)
532             self.save_path = jsonfile['save_path']
533             self.time_end = time.time()
534             m, s = divmod(int(self.time_end-self.time_start), 60)
535             h, m = divmod(m, 60)
536             self.recording_time =
537                 ↪ str(h)+'时'+str(m)+'分'+str(s)+'秒'
538             self.statusbar.showMessage('当前样本编号: ' +
539                 ↪ self.lineEdit.text() + '    保存位置: ' +
540                 ↪ self.save_path + '    录制时间: ' +
541                 ↪ self.recording_time)
542             if self._start_running == True:
543                 self.qttimer.stop()
544     def startRe(self):
545         print(self.voice_index)
546         if self._start_running:
547             self._start_running =False
548             self._begin = True
549             with open("config/db.json", "r", encoding='UTF-8')
550                 ↪ as dbfile_r:
551                     jsonfile = json.load(dbfile_r)
552             self.voice_format = jsonfile["voice_format"]
553             self.video_format = jsonfile["video_format"]
554             #需要显示的有: 病人ID, 当前保存路径,
555             ↪ 已经录制多长时间, 在加一个录制中。
556             #已录制多长时间需要动态显示, 每秒更新一次状态栏,
557             ↪ 只在录制视频的时候显示。
558             #什么情况下会录制视频? _begin == True ,上面写了,
559             ↪ 不等于-1就能录制视频。

```

```

550         if self.voice_index != -1 or self.eye_camera != -1
↪         or self.hd_camera != -1 or self.face_camera !=
↪         -1:#self.voice_index! =-1, 则一定会录制音频。
↪         如果有其他设备不为-1则会录制视频,
↪         只要有录制情况就显示该消息
551         self.time_start = time.time()
552         self.record_judge = True #表示是否在录制中。
↪         该变量只与qtimer有关,
↪         该变量为True则一定有QTimer ()
↪         ,当然线程终止也会赋值为False,
↪         表示结束录制了。
553         self.qttimer = QTimer()
554         self.qttimer.start(200)
555         self.qttimer.timeout.connect(self.updatemessa_
↪         ge)
556         threading._start_new_thread(self.save_hd_video,
↪         ())
557         threading._start_new_thread(self.save_face_video,
↪         ())
558         threading._start_new_thread(self.save_eye_video,
↪         ())
559         if self.voice_index != -1:
560             threading._start_new_thread(self.win_4, ())
561         else:
562             pass
563     def endRe(self):
564         if self._start_running==False:
565             if self.record_judge == True:# 只有点击
566                 self.qttimer.stop()
567                 self.record_judge = False
568                 self.statusbar.showMessage('录制结束,
↪                 文件已保存至: ' + self.save_path + '
↪                 录制时长: ' + self.recording_time)
569                 self._start_running=True
570                 self._begin =False
571                 if self.voice_index != -1:
572                     self._voice_get = False
573                     self.voice_save()

```

```

574         self.statusbar.showMessage('录制结束，
↪ 文件已保存至: '+self.save_path+' 录制时长:
↪ '+self.recording_time)
575     else:
576         pass
577     def closeEvent(self,event):
578         # 关闭所有线程，防止不能退出
579         # 如果没有运行进程终止，则不能退出
580         if self._init_running == False:
581             QMessageBox.information(self,'警告','进程未中止，
↪ 请中止进程再关闭窗口! ')
582             event.ignore()
583             print('closeevent')
584     def stop(self):
585         if self.record_judge == True:
586             QMessageBox.information(self,'警告','正在录制中，
↪ 请结束录制后再点击停用视频设备! ')
587         else:
588             self._win1_running = False
589             self._win2_running = False
590             self._win3_running = False
591             self._init_running = True
592             print('stop','所有通道均已断开')
593             self.voice_image.updateImage(img_no_voice)
594             self.statusbar.showMessage('所有通道均已断开')
595     def f_p(self):
596         #点进程终止的时候录音进程可能不会终止，而视频录制会结束，
↪ 必须再点结束录制才能结束-----先不管它，
↪ 假设先用不到它。-----还是先解决它吧。
597         #进程中止前判断是否在录制中，若在录制中，需要点击录制结束
598         if self.record_judge == True:
599             QMessageBox.information(self,'警告','正在录制中，
↪ 请结束录制后再点击进程中止! ')
600         else:
601             self._win1_running = False
602             self._win2_running = False
603             self._win3_running = False
604             self._init_running = True
605             if self.record_judge == True:

```

```

606         self._begin = False
607         self._start_running = True
608         self._voice_get = False
609         self.voice_save()
610         self.qttimer.stop()
611         self.record_judge = False
612         self._voice_get = True
613         self._begin = False
614         print('stop', '所有进程均已中止')
615         self.statusbar.showMessage('所有进程均已中止')
616         self.voice_image.updateImage(img_no_voice)
617     def about(self):
618         QMessageBox.about(self, " 面部表型特征提取系统",
619                             "<p> 山东省精神卫生中心 </p>")
620     def updateImages(self):
621         self.calculateProcessed()
622         self.calculateOriginal()
623     def createButtons(self):
624         self.initButton.clicked.connect(self.initfrom)
625         self.pushButton.clicked.connect(self.stop)
626         self.shotButton.clicked.connect(self.f_p)
627         self.actionAbout.triggered.connect(self.about)
628         self.startButton.clicked.connect(self.startRe)
629         self.endButton.clicked.connect(self.endRe)
630         self.action12.triggered.connect(self.child_CameraVoic_
        ↪ e_setting)
        ↪ #子窗口配置摄像头和麦克风
631         self.actionse.triggered.connect(self.Child_File_Setti_
        ↪ ng)#选择文件路径
632     def setBackground(self):
633         cv_img_rgb = img_no_camera
634         self.original1_image.updateImage(cv_img_rgb)
635         self.original2_image.updateImage(cv_img_rgb)
636         self.eye_image.updateImage(cv_img_rgb)
637         self.voice_image.updateImage(img_no_voice)

```

1.15 CvPyGui/ui/___init___py

1

1.16 CvPyGui/ui/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)
28         self.layoutWidget.setGeometry(QtCore.QRect(14, 5, 361,
29             ↳ 71))
30         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("horizontalLayout_2")  
            ↳ 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_path_label")  
            ↳ th_label")  
35         self.horizontalLayout_2.addWidget(self.show_file_path_label)  
            ↳ _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.QSizePolicy.Minimum,  
            ↳ icycle.Minimum,  
            ↳ QtWidgets.QSizePolicy.Preferred)  
43         sizePolicy.setHorizontalStretch(0)  
44         sizePolicy.setVerticalStretch(0)  
45         sizePolicy.setHeightForWidth(self.file_path_label.sizePolicy().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(QRect(40, 130,
        ↪ 291, 22))
78         self.layoutWidget2.setObjectName("layoutWidget2")
79         self.horizontalLayout_4 =
        ↪ QtWidgets.QHBoxLayout(self.layoutWidget2)
80         self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
81         self.horizontalLayout_4.setObjectName("horizontalLayo_
        ↪ ut_4")
82         self.label_2 = QtWidgets.QLabel(self.layoutWidget2)
83         self.label_2.setObjectName("label_2")
84         self.horizontalLayout_4.addWidget(self.label_2)
85         self.radioButton_mp4 =
        ↪ QtWidgets.QRadioButton(self.layoutWidget2)
86         self.radioButton_mp4.setObjectName("radioButton_mp4")
87         self.horizontalLayout_4.addWidget(self.radioButton_mp
        ↪ 4)
88         self.radioButton_avi =
        ↪ QtWidgets.QRadioButton(self.layoutWidget2)
89         self.radioButton_avi.setObjectName("radioButton_avi")
90         self.horizontalLayout_4.addWidget(self.radioButton_av
        ↪ i)
91
92         self.retranslateUi(Form)
93         QtCore.QMetaObject.connectSlotsByName(Form)
94
95     def retranslateUi(self, Form):
96         _translate = QtCore.QCoreApplication.translate
97         Form.setWindowTitle(_translate("Form", " 初始化配置"))
98         self.file_ok.setText(_translate("Form", " 确认"))
99         self.label.setText(_translate("Form",
        ↪ " 音视频格式选择"))
100        self.show_file_path_label.setText(_translate("Form",
        ↪ " 当前文件保存路径:"))
101        self.file_path_label.setText(_translate("Form", "C:"))
102        self.choice_file_path_button.setText(_translate("Form",
        ↪ " ,
        ↪ 选择文件保存路径"))
103        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_())

```

1.17 CvPyGui/ui/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 hsizeType="Fixed" vsizeType="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 class="QWidget" name="layoutWidget">
118     <property name="geometry">
119         <rect>
120             <x>40</x>
121             <y>180</y>
122             <width>291</width>
123             <height>22</height>
124         </rect>
125     </property>
126     <layout class="QHBoxLayout" name="horizontalLayout_3">
127         <item>
128             <widget class="QLabel" name="label_3">
129                 <property name="text">
130                     <string>音频格式</string>
131                 </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>
```

1.18 CvPyGui/ui/child_test.py

```
1 from PyQt5 import QtCore, QtGui, QtWidgets
2 class Ui_Form(object):
3     def setupUi(self, Form):
4         Form.setObjectName("Form")
5         Form.resize(400, 310)
6         self.tabWidget = QtWidgets.QTabWidget(Form)
7         self.tabWidget.setGeometry(QtCore.QRect(10, 10, 381,
8             ↪ 261))
9         self.tabWidget.setObjectName("tabWidget")
10        self.tab = QtWidgets.QWidget()
11        self.tab.setObjectName("tab")
12        self.layoutWidget = QtWidgets.QWidget(self.tab)
```



```
12         self.layoutWidget.setGeometry(QRect(0, 10, 371,
13         ↪ 211))
14         self.layoutWidget.setObjectName("layoutWidget")
15         self.horizontalLayout_2 =
16         ↪ QtWidgets.QHBoxLayout(self.layoutWidget)
17         self.horizontalLayout_2.setContentsMargins(0, 0, 0, 0)
18         self.horizontalLayout_2.setObjectName("horizontalLayo_
19         ↪ ut_2")
20         spacerItem = QtWidgets.QSpacerItem(40, 20,
21         ↪ QtWidgets.QSizePolicy.Expanding,
22         ↪ QtWidgets.QSizePolicy.Minimum)
23         self.horizontalLayout_2.addItem(spacerItem)
24         self.verticalLayout_2 = QtWidgets.QVBoxLayout()
25         self.verticalLayout_2.setObjectName("verticalLayout_2_
26         ↪ ")
27         self.verticalLayout = QtWidgets.QVBoxLayout()
28         self.verticalLayout.setObjectName("verticalLayout")
29         self.horizontalLayout = QtWidgets.QHBoxLayout()
30         self.horizontalLayout.setObjectName("horizontalLayout_
31         ↪ ")
32         spacerItem1 = QtWidgets.QSpacerItem(30, 20,
33         ↪ QtWidgets.QSizePolicy.Maximum,
34         ↪ QtWidgets.QSizePolicy.Minimum)
35         self.horizontalLayout.addItem(spacerItem1)
36         self.tab1_combobox =
37         ↪ QtWidgets.QComboBox(self.layoutWidget)
38         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
39         ↪ icky.MinimumExpanding,
40         ↪ QtWidgets.QSizePolicy.Fixed)
41         sizePolicy.setHorizontalStretch(0)
42         sizePolicy.setVerticalStretch(0)
43         sizePolicy.setHeightForWidth(self.tab1_combobox.sizeP_
44         ↪ olicy().hasHeightForWidth())
45         self.tab1_combobox.setSizePolicy(sizePolicy)
46         self.tab1_combobox.setObjectName("tab1_combobox")
47         self.horizontalLayout.addWidget(self.tab1_combobox)
48         spacerItem2 = QtWidgets.QSpacerItem(30, 20,
49         ↪ QtWidgets.QSizePolicy.Maximum,
50         ↪ QtWidgets.QSizePolicy.Minimum)
```

```

36         self.horizontalLayout.addItem(spacerItem2)
37         self.verticalLayout.addLayout(self.horizontalLayout)
38         spacerItem3 = QtWidgets.QSpacerItem(40, 20,
39         ↪ QtWidgets.QSizePolicy.Expanding,
40         ↪ QtWidgets.QSizePolicy.Minimum)
41         self.verticalLayout.addItem(spacerItem3)
42         self.verticalLayout_2.addLayout(self.verticalLayout)
43         self.tab1_label = QtWidgets.QLabel(self.layoutWidget)
44         sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
45         ↪ icy.Ignored,
46         ↪ QtWidgets.QSizePolicy.Fixed)
47         sizePolicy.setHorizontalStretch(0)
48         sizePolicy.setVerticalStretch(0)
49         sizePolicy.setHeightForWidth(self.tab1_label.sizePoli_
50         ↪ cy().hasHeightForWidth())
51         self.tab1_label.setSizePolicy(sizePolicy)
52         self.tab1_label.setMinimumSize(QtCore.QSize(256, 144))
53         self.tab1_label.setMaximumSize(QtCore.QSize(256, 144))
54         self.tab1_label.setAlignment(QtCore.Qt.AlignCenter)
55         self.tab1_label.setObjectName("tab1_label")
56         self.verticalLayout_2.addWidget(self.tab1_label)
57         self.horizontalLayout_2.addLayout(self.verticalLayout_
58         ↪ _2)
59         spacerItem4 = QtWidgets.QSpacerItem(40, 20,
60         ↪ QtWidgets.QSizePolicy.Expanding,
61         ↪ QtWidgets.QSizePolicy.Minimum)
62         self.horizontalLayout_2.addItem(spacerItem4)
63         self.tabWidget.addTab(self.tab, "")
64         self.tab_2 = QtWidgets.QWidget()
65         self.tab_2.setObjectName("tab_2")
66         self.layoutWidget_2 = QtWidgets.QWidget(self.tab_2)
67         self.layoutWidget_2.setGeometry(QtCore.QRect(0, 10,
68         ↪ 371, 211))
69         self.layoutWidget_2.setObjectName("layoutWidget_2")
70         self.horizontalLayout_7 =
71         ↪ QtWidgets.QHBoxLayout(self.layoutWidget_2)
72         self.horizontalLayout_7.setContentsMargins(0, 0, 0, 0)
73         self.horizontalLayout_7.setObjectName("horizontalLayo_
74         ↪ ut_7")

```

```
64         spacerItem5 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
65     self.horizontalLayout_7.addItem(spacerItem5)
66     self.verticalLayout_5 = QtWidgets.QVBoxLayout()
67     self.verticalLayout_5.setObjectName("verticalLayout_5")
        ↪ ")
68     self.verticalLayout_6 = QtWidgets.QVBoxLayout()
69     self.verticalLayout_6.setObjectName("verticalLayout_6")
        ↪ ")
70     self.horizontalLayout_8 = QtWidgets.QHBoxLayout()
71     self.horizontalLayout_8.setObjectName("horizontalLayo
        ↪ ut_8")
72     spacerItem6 = QtWidgets.QSpacerItem(30, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
73     self.horizontalLayout_8.addItem(spacerItem6)
74     self.tab1_combobox_3 =
        ↪ QtWidgets.QComboBox(self.layoutWidget_2)
75     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol
        ↪ icy.MinimumExpanding,
        ↪ QtWidgets.QSizePolicy.Fixed)
76     sizePolicy.setHorizontalStretch(0)
77     sizePolicy.setVerticalStretch(0)
78     sizePolicy.setHeightForWidth(self.tab1_combobox_3.siz
        ↪ ePolicy().hasHeightForWidth())
79     self.tab1_combobox_3.setSizePolicy(sizePolicy)
80     self.tab1_combobox_3.setObjectName("tab1_combobox_3")
81     self.horizontalLayout_8.addWidget(self.tab1_combobox_
        ↪ 3)
82     spacerItem7 = QtWidgets.QSpacerItem(30, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
83     self.horizontalLayout_8.addItem(spacerItem7)
84     self.verticalLayout_6.addLayout(self.horizontalLayout_
        ↪ _8)
85     spacerItem8 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
```

```

86         self.verticalLayout_6.addItem(spacerItem8)
87         self.verticalLayout_5.addLayout(self.verticalLayout_6)
88         self.tab1_label_3 =
89             ↳ QtWidgets.QLabel(self.layoutWidget_2)
90             sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
91             ↳ icy.Ignored,
92             ↳ QtWidgets.QSizePolicy.Fixed)
93             sizePolicy.setHorizontalStretch(0)
94             sizePolicy.setVerticalStretch(0)
95             sizePolicy.setHeightForWidth(self.tab1_label_3.sizePo_
96             ↳ licy().hasHeightForWidth())
97         self.tab1_label_3.setSizePolicy(sizePolicy)
98         self.tab1_label_3.setMinimumSize(QtCore.QSize(256,
99             ↳ 144))
100        self.tab1_label_3.setMaximumSize(QtCore.QSize(256,
101            ↳ 144))
102        self.tab1_label_3.setAlignment(QtCore.Qt.AlignCenter)
103        self.tab1_label_3.setObjectName("tab1_label_3")
104        self.verticalLayout_5.addWidget(self.tab1_label_3)
105        self.horizontalLayout_7.addLayout(self.verticalLayout_
106            ↳ _5)
107        spacerItem9 = QtWidgets.QSpacerItem(40, 20,
108            ↳ QtWidgets.QSizePolicy.Expanding,
109            ↳ QtWidgets.QSizePolicy.Minimum)
110        self.horizontalLayout_7.addItem(spacerItem9)
111        # self.tabWidget.addTab(self.tab_2, "")
112        self.tab_3 = QtWidgets.QWidget()
113        self.tab_3.setObjectName("tab_3")
114        self.layoutWidget_3 = QtWidgets.QWidget(self.tab_3)
115        self.layoutWidget_3.setGeometry(QtCore.QRect(0, 10,
116            ↳ 371, 211))
117        self.layoutWidget_3.setObjectName("layoutWidget_3")
118        self.horizontalLayout_4 =
119            ↳ QtWidgets.QHBoxLayout(self.layoutWidget_3)
120        self.horizontalLayout_4.setContentsMargins(0, 0, 0, 0)
121        self.horizontalLayout_4.setObjectName("horizontalLayo_
122            ↳ ut_4")

```

```
111         spacerItem10 = QtWidgets.QSpacerItem(40, 20,  
        ↪ QtWidgets.QSizePolicy.Expanding,  
        ↪ QtWidgets.QSizePolicy.Minimum)  
112     self.horizontalLayout_4.addItem(spacerItem10)  
113     self.verticalLayout_7 = QtWidgets.QVBoxLayout()  
114     self.verticalLayout_7.setObjectName("verticalLayout_7"  
        ↪ ")  
115     self.verticalLayout_8 = QtWidgets.QVBoxLayout()  
116     self.verticalLayout_8.setObjectName("verticalLayout_8"  
        ↪ ")  
117     self.horizontalLayout_9 = QtWidgets.QHBoxLayout()  
118     self.horizontalLayout_9.setObjectName("horizontalLayo  
        ↪ ut_9")  
119     spacerItem11 = QtWidgets.QSpacerItem(30, 20,  
        ↪ QtWidgets.QSizePolicy.Maximum,  
        ↪ QtWidgets.QSizePolicy.Minimum)  
120     self.horizontalLayout_9.addItem(spacerItem11)  
121     self.tab1_combobox_4 =  
        ↪ QtWidgets.QComboBox(self.layoutWidget_3)  
122     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol  
        ↪ icy.MinimumExpanding,  
        ↪ QtWidgets.QSizePolicy.Fixed)  
123     sizePolicy.setHorizontalStretch(0)  
124     sizePolicy.setVerticalStretch(0)  
125     sizePolicy.setHeightForWidth(self.tab1_combobox_4.siz  
        ↪ ePolicy().hasHeightForWidth())  
126     self.tab1_combobox_4.setSizePolicy(sizePolicy)  
127     self.tab1_combobox_4.setObjectName("tab1_combobox_4")  
128     self.horizontalLayout_9.addWidget(self.tab1_combobox_  
        ↪ 4)  
129     spacerItem12 = QtWidgets.QSpacerItem(30, 20,  
        ↪ QtWidgets.QSizePolicy.Maximum,  
        ↪ QtWidgets.QSizePolicy.Minimum)  
130     self.horizontalLayout_9.addItem(spacerItem12)  
131     self.verticalLayout_8.addLayout(self.horizontalLayout_  
        ↪ _9)  
132     spacerItem13 = QtWidgets.QSpacerItem(40, 20,  
        ↪ QtWidgets.QSizePolicy.Expanding,  
        ↪ QtWidgets.QSizePolicy.Minimum)
```

```

133     self.verticalLayout_8.addItem(spacerItem13)
134     self.verticalLayout_7.addLayout(self.verticalLayout_8)
135     self.tab1_label_4 =
        ↳ QtWidgets.QLabel(self.layoutWidget_3)
136     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePol_
        ↳ icy.Ignored,
        ↳ QtWidgets.QSizePolicy.Fixed)
137     sizePolicy.setHorizontalStretch(0)
138     sizePolicy.setVerticalStretch(0)
139     sizePolicy.setHeightForWidth(self.tab1_label_4.sizePo_
        ↳ licy().hasHeightForWidth())
140     self.tab1_label_4.setSizePolicy(sizePolicy)
141     self.tab1_label_4.setMinimumSize(QtCore.QSize(256,
        ↳ 144))
142     self.tab1_label_4.setMaximumSize(QtCore.QSize(256,
        ↳ 144))
143     self.tab1_label_4.setAlignment(QtCore.Qt.AlignCenter)
144     self.tab1_label_4.setObjectName("tab1_label_4")
145     self.verticalLayout_7.addWidget(self.tab1_label_4)
146     self.horizontalLayout_4.addLayout(self.verticalLayout_
        ↳ _7)
147     spacerItem14 = QtWidgets.QSpacerItem(40, 20,
        ↳ QtWidgets.QSizePolicy.Expanding,
        ↳ QtWidgets.QSizePolicy.Minimum)
148     self.horizontalLayout_4.addItem(spacerItem14)
149     # self.tabWidget.addTab(self.tab_3, "")
150     self.tab_4 = QtWidgets.QWidget()
151     self.tab_4.setObjectName("tab_4")
152     self.layoutWidget1 = QtWidgets.QWidget(self.tab_4)
153     self.layoutWidget1.setGeometry(QtCore.QRect(0, 10,
        ↳ 371, 211))
154     self.layoutWidget1.setObjectName("layoutWidget1")
155     self.horizontalLayout_13 =
        ↳ QtWidgets.QHBoxLayout(self.layoutWidget1)
156     self.horizontalLayout_13.setContentsMargins(0, 0, 0,
        ↳ 0)
157     self.horizontalLayout_13.setObjectName("horizontalLay_
        ↳ out_13")

```

```
158         spacerItem15 = QtWidgets.QSpacerItem(40, 40,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
159     self.horizontalLayout_13.addItem(spacerItem15)
160     self.verticalLayout_13 = QtWidgets.QVBoxLayout()
161     self.verticalLayout_13.setObjectName("verticalLayout_13")
        ↪ 13")
162     self.verticalLayout_14 = QtWidgets.QVBoxLayout()
163     self.verticalLayout_14.setObjectName("verticalLayout_14")
        ↪ 14")
164     self.horizontalLayout_14 = QtWidgets.QHBoxLayout()
165     self.horizontalLayout_14.setObjectName("horizontalLayout_14")
        ↪ out_14")
166     spacerItem16 = QtWidgets.QSpacerItem(31, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
167     self.horizontalLayout_14.addItem(spacerItem16)
168     self.tab1_combobox_7 =
        ↪ QtWidgets.QComboBox(self.layoutWidget1)
169     sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.MinimumExpanding,
        ↪ QtWidgets.QSizePolicy.Fixed)
170     sizePolicy.setHorizontalStretch(0)
171     sizePolicy.setVerticalStretch(0)
172     sizePolicy.setHeightForWidth(self.tab1_combobox_7.sizePolicy().hasHeightForWidth())
173     self.tab1_combobox_7.setSizePolicy(sizePolicy)
174     self.tab1_combobox_7.setMinimumSize(QtCore.QSize(160, 0))
175     self.tab1_combobox_7.setObjectName("tab1_combobox_7")
176     self.horizontalLayout_14.addWidget(self.tab1_combobox_7)
        ↪ _7)
177     spacerItem17 = QtWidgets.QSpacerItem(31, 20,
        ↪ QtWidgets.QSizePolicy.Maximum,
        ↪ QtWidgets.QSizePolicy.Minimum)
178     self.horizontalLayout_14.addItem(spacerItem17)
179     self.verticalLayout_14.addLayout(self.horizontalLayout_14)
        ↪ t_14)
```

```

180         spacerItem18 = QtWidgets.QSpacerItem(256, 55,
        ↪ QtWidgets.QSizePolicy.Minimum,
        ↪ QtWidgets.QSizePolicy.Minimum)
181     self.verticalLayout_14.addItem(spacerItem18)
182     self.verticalLayout_13.addLayout(self.verticalLayout_1
        ↪ 14)
183     self.progressBar =
        ↪ QtWidgets.QProgressBar(self.layoutWidget1)
184     self.progressBar.setProperty("value", 0)
185     self.progressBar.setTextVisible(False)
186     self.progressBar.setObjectName("progressBar")
187     self.verticalLayout_13.addWidget(self.progressBar)
188     spacerItem19 = QtWidgets.QSpacerItem(40, 60,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
189     self.verticalLayout_13.addItem(spacerItem19)
190     self.horizontalLayout_13.addLayout(self.verticalLayout_1
        ↪ t_13)
191     spacerItem20 = QtWidgets.QSpacerItem(40, 20,
        ↪ QtWidgets.QSizePolicy.Expanding,
        ↪ QtWidgets.QSizePolicy.Minimum)
192     self.horizontalLayout_13.addItem(spacerItem20)
193     self.tabWidget.addTab(self.tab_4, "")
194     self.OK = QtWidgets.QPushButton(Form)
195     self.OK.setGeometry(QtCore.QRect(260, 280, 111, 23))
196     self.OK.setObjectName("OK")
197     self.retranslateUi(Form)
198     self.tabWidget.setCurrentIndex(0)
199     QtCore.QMetaObject.connectSlotsByName(Form)
200     def retranslateUi(self, Form):
201         _translate = QtCore.QCoreApplication.translate
202         Form.setWindowTitle(_translate("Form",
        ↪ "音视频设备设置"))
203         self.tab1_label.setText(_translate("Form",
        ↪ "TextLabel"))
204         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪ .tab), _translate("Form",
        ↪ "视频源"))

```



```

205         self.tab1_label_3.setText(_translate("Form",
        ↪ "TextLabel"))
206         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪ .tab_2), _translate("Form",
        ↪ "面部"))
207         self.tab1_label_4.setText(_translate("Form",
        ↪ "TextLabel"))
208         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪ .tab_3), _translate("Form",
        ↪ "眼睛"))
209         self.tabWidget.setTabText(self.tabWidget.indexOf(self
        ↪ .tab_4), _translate("Form",
        ↪ "音频源"))
210         self.OK.setText(_translate("Form", " 确认"))

```

1.19 CvPyGui/ui/gui.py

```

1  # -*- coding: utf-8 -*-
2  # Form implementation generated from reading ui file 'gui.ui'
3  #
4  # Created by: PyQt5 UI code generator 5.9.2
5  #
6  # WARNING! All changes made in this file will be lost!
7  from PyQt5 import QtCore, QtGui, QtWidgets
8  class Ui_MainWindow(object):
9      def setupUi(self, MainWindow):
10         MainWindow.setObjectName("MainWindow")
11         MainWindow.resize(830, 578)
12         icon = QtGui.QIcon()
13         icon.addPixmap(QtGui.QPixmap("icon.ico"),
        ↪ QtGui.QIcon.Normal, QtGui.QIcon.Off)
14         MainWindow.setWindowIcon(icon)
15         self.centralwidget = QtWidgets.QWidget(MainWindow)
16         self.centralwidget.setObjectName("centralwidget")
17         self.verticalLayout_4 =
        ↪ QtWidgets.QVBoxLayout(self.centralwidget)
18         self.verticalLayout_4.setObjectName("verticalLayout_4
        ↪ ")

```

```
19         self.h_img_layout = QtWidgets.QHBoxLayout()
20         self.h_img_layout.setObjectName("h_img_layout")
21         self.original_img_v_layout = QtWidgets.QVBoxLayout()
22         self.original_img_v_layout.setObjectName("original_img_v_l
    ↪ _layout")
23         spacerItem = QtWidgets.QSpacerItem(20, 40,
    ↪ QtWidgets.QSizePolicy.Minimum,
    ↪ QtWidgets.QSizePolicy.Expanding)
24         self.original_img_v_layout.addItem(spacerItem)
25         self.original_name_label =
    ↪ QtWidgets.QLabel(self.centralwidget)
26         self.original_name_label.setScaledContents(False)
27         self.original_name_label.setAlignment(QtCore.Qt.AlignCe
    ↪ nter)
28         self.original_name_label.setObjectName("original_name_l
    ↪ bl")
29         self.original_img_v_layout.addWidget(self.original_name_
    ↪ label)
30         spacerItem1 = QtWidgets.QSpacerItem(20, 40,
    ↪ QtWidgets.QSizePolicy.Minimum,
    ↪ QtWidgets.QSizePolicy.Expanding)
31         self.original_img_v_layout.addItem(spacerItem1)
32         self.original_frame_label =
    ↪ QtWidgets.QLabel(self.centralwidget)
33         self.original_frame_label.setMinimumSize(QtCore.QSize(4
    ↪ 00,
    ↪ 240))
34         self.original_frame_label.setAlignment(QtCore.Qt.AlignC
    ↪ enter)
35         self.original_frame_label.setObjectName("original_frame
    ↪ _label")
36         self.original_img_v_layout.addWidget(self.original_frame
    ↪ _label)
37         spacerItem2 = QtWidgets.QSpacerItem(20, 40,
    ↪ QtWidgets.QSizePolicy.Minimum,
    ↪ QtWidgets.QSizePolicy.Expanding)
38         self.original_img_v_layout.addItem(spacerItem2)
39         self.h_img_layout.addLayout(self.original_img_v_layout)
40         self.processed_img_v_layout = QtWidgets.QVBoxLayout()
```

```
41     self.processed_img_v_layout.setObjectName("processed_img_v_
    ↳ _layout")
42     spacerItem3 = QtWidgets.QSpacerItem(20, 40,
    ↳ QtWidgets.QSizePolicy.Minimum,
    ↳ QtWidgets.QSizePolicy.Expanding)
43     self.processed_img_v_layout.addItem(spacerItem3)
44     self.processed_name_label =
    ↳ QtWidgets.QLabel(self.centralwidget)
45     self.processed_name_label.setAlignment(QtCore.Qt.AlignC
    ↳ enter)
46     self.processed_name_label.setObjectName("processed_name_
    ↳ _label")
47     self.processed_img_v_layout.addWidget(self.processed_nam
    ↳ e_label)
48     spacerItem4 = QtWidgets.QSpacerItem(20, 40,
    ↳ QtWidgets.QSizePolicy.Minimum,
    ↳ QtWidgets.QSizePolicy.Expanding)
49     self.processed_img_v_layout.addItem(spacerItem4)
50     self.processed_frame_label =
    ↳ QtWidgets.QLabel(self.centralwidget)
51     self.processed_frame_label.setMinimumSize(QtCore.QSize(
    ↳ 400,
    ↳ 240))
52     self.processed_frame_label.setBaseSize(QtCore.QSize(0,
    ↳ 0))
53     self.processed_frame_label.setText("")
54     self.processed_frame_label.setAlignment(QtCore.Qt.Align
    ↳ Center)
55     self.processed_frame_label.setObjectName("processed_fra
    ↳ me_label")
56     self.processed_img_v_layout.addWidget(self.processed_fra
    ↳ me_label)
57     spacerItem5 = QtWidgets.QSpacerItem(20, 40,
    ↳ QtWidgets.QSizePolicy.Minimum,
    ↳ QtWidgets.QSizePolicy.Expanding)
58     self.processed_img_v_layout.addItem(spacerItem5)
59     self.h_img_layout.addLayout(self.processed_img_v_layout)
60     self.verticalLayout_4.addLayout(self.h_img_layout)
61     self.horizontalLayout = QtWidgets.QHBoxLayout()
```

```
62         self.horizontalLayout.setObjectName("horizontalLayout_1")
63         self.verticalLayout_2 = QtWidgets.QVBoxLayout()
64         self.verticalLayout_2.setObjectName("verticalLayout_2")
65         self.label_3 = QtWidgets.QLabel(self.centralwidget)
66         self.label_3.setAlignment(Qt.AlignCenter)
67         self.label_3.setObjectName("label_3")
68         self.verticalLayout_2.addWidget(self.label_3)
69         self.eye_lbl = QtWidgets.QLabel(self.centralwidget)
70         self.eye_lbl.setMinimumSize(QtCore.QSize(400, 240))
71         self.eye_lbl.setText("")
72         self.eye_lbl.setAlignment(Qt.AlignCenter)
73         self.eye_lbl.setObjectName("eye_lbl")
74         self.verticalLayout_2.addWidget(self.eye_lbl)
75         self.horizontalLayout.addLayout(self.verticalLayout_2)
76         self.verticalLayout = QtWidgets.QVBoxLayout()
77         self.verticalLayout.setObjectName("verticalLayout")
78         self.label_2 = QtWidgets.QLabel(self.centralwidget)
79         self.label_2.setAlignment(Qt.AlignCenter)
80         self.label_2.setObjectName("label_2")
81         self.verticalLayout.addWidget(self.label_2)
82         self.voice_lbl = QtWidgets.QLabel(self.centralwidget)
83         self.voice_lbl.setMinimumSize(QtCore.QSize(400, 240))
84         self.voice_lbl.setText("")
85         self.voice_lbl.setAlignment(Qt.AlignCenter)
86         self.voice_lbl.setObjectName("voice_lbl")
87         self.verticalLayout.addWidget(self.voice_lbl)
88         self.horizontalLayout.addLayout(self.verticalLayout)
89         self.verticalLayout_4.addLayout(self.horizontalLayout)
90         self.h_btn_layout = QtWidgets.QHBoxLayout()
91         self.h_btn_layout.setObjectName("h_btn_layout")
92         self.initButton =
93             QtWidgets.QPushButton(self.centralwidget)
94         self.initButton.setObjectName("initButton")
95         self.h_btn_layout.addWidget(self.initButton)
96         self.pushButton =
97             QtWidgets.QPushButton(self.centralwidget)
98         self.pushButton.setObjectName("pushButton")
```

```
97         self.h_btn_layout.addWidget(self.pushButton)
98         spacerItem6 = QtWidgets.QSpacerItem(40, 20,
99         ↪ QtWidgets.QSizePolicy.Expanding,
100         ↪ QtWidgets.QSizePolicy.Minimum)
101         self.h_btn_layout.addItem(spacerItem6)
102         self.label = QtWidgets.QLabel(self.centralwidget)
103         self.label.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.
104         ↪ t.AlignTrailing|QtCore.Qt.AlignVCenter)
105         self.label.setObjectName("label")
106         self.h_btn_layout.addWidget(self.label)
107         self.lineEdit =
108         ↪ QtWidgets.QLineEdit(self.centralwidget)
109         self.lineEdit.setObjectName("lineEdit")
110         self.h_btn_layout.addWidget(self.lineEdit)
111         self.startButton =
112         ↪ QtWidgets.QPushButton(self.centralwidget)
113         self.startButton.setObjectName("startButton")
114         self.h_btn_layout.addWidget(self.startButton)
115         self.endButton =
116         ↪ QtWidgets.QPushButton(self.centralwidget)
117         self.endButton.setObjectName("endButton")
118         self.h_btn_layout.addWidget(self.endButton)
119         self.shotButton =
120         ↪ QtWidgets.QPushButton(self.centralwidget)
121         self.shotButton.setObjectName("shotButton")
122         self.h_btn_layout.addWidget(self.shotButton)
123         self.verticalLayout_4.addLayout(self.h_btn_layout)
124         self.v_filters_layout = QtWidgets.QVBoxLayout()
125         self.v_filters_layout.setObjectName("v_filters_layout")
126         self.verticalLayout_4.addLayout(self.v_filters_layout)
127         MainWindow.setCentralWidget(self.centralwidget)
128         self.menubar = QtWidgets.QMenuBar(MainWindow)
129         self.menubar.setGeometry(QtCore.QRect(0, 0, 1035, 23))
130         self.menubar.setObjectName("menubar")
131         self.menuHelp = QtWidgets.QMenu(self.menubar)
132         self.menuHelp.setObjectName("menuHelp")
133         self.menu = QtWidgets.QMenu(self.menubar)
134         self.menu.setObjectName("menu")
135         self.menu_2 = QtWidgets.QMenu(self.menubar)
```

```
129         self.menu_2.setObjectName("menu_2")
130         MainWindow.setMenuBar(self.menubar)
131         self.statusbar = QtWidgets.QStatusBar(MainWindow)
132         self.statusbar.setMinimumSize(QtCore.QSize(500, 30))
133         self.statusbar.setObjectName("statusbar")
134         MainWindow.setStatusBar(self.statusbar)
135         self.actionOpen_image = QtWidgets.QAction(MainWindow)
136         self.actionOpen_image.setObjectName("actionOpen_image")
137         self.actionSave_original_image =
138             QtWidgets.QAction(MainWindow)
139         self.actionSave_original_image.setObjectName("actionS_
140             ave_original_image")
141         self.actionSave_processed_image =
142             QtWidgets.QAction(MainWindow)
143         self.actionSave_processed_image.setObjectName("action_
144             Save_processed_image")
145         self.actionExit = QtWidgets.QAction(MainWindow)
146         self.actionExit.setObjectName("actionExit")
147         self.actionLicense = QtWidgets.QAction(MainWindow)
148         self.actionLicense.setObjectName("actionLicense")
149         self.actionAbout = QtWidgets.QAction(MainWindow)
150         self.actionAbout.setObjectName("actionAbout")
151         self.actionmanual = QtWidgets.QAction(MainWindow)
152         self.actionmanual.setObjectName("actionmanual")
153         self.action12 = QtWidgets.QAction(MainWindow)
154         self.action12.setObjectName("action12")
155         self.actionse = QtWidgets.QAction(MainWindow)
156         self.actionse.setObjectName("actionse")
157         self.menuHelp.addAction(self.actionmanual)
158         self.menuHelp.addAction(self.actionAbout)
159         self.menu_2.addAction(self.action12)
160         self.menu_2.addAction(self.actionse)
161         self.menubar.addAction(self.menu.menuAction())
162         self.menubar.addAction(self.menu_2.menuAction())
163         self.menubar.addAction(self.menuHelp.menuAction())
164         self.retranslateUi(MainWindow)
165         QtCore.QMetaObject.connectSlotsByName(MainWindow)
166     def retranslateUi(self, MainWindow):
```

```

163         _translate = QtCore.QCoreApplication.translate
164         MainWindow.setWindowTitle(_translate("MainWindow",
        ↳ "肢体表型特征采集系统---山东省精神卫生中心"))
165         self.original_name_lbl.setText(_translate("MainWindow",
        ↳ " ",
        ↳ "原始图像"))
166         self.processed_name_lbl.setText(_translate("MainWindo",
        ↳ "w",
        ↳ "肢体特征"))
167         self.label_3.setText(_translate("MainWindow",
        ↳ "标记图像"))
168         self.label_2.setText(_translate("MainWindow",
        ↳ "肢体轮廓"))
169         self.initButton.setText(_translate("MainWindow",
        ↳ "检测视频设备"))
170         self.pushButton.setText(_translate("MainWindow",
        ↳ "停用视频设备"))
171         self.label.setText(_translate("MainWindow", "样本编号:
        ↳ "))
172         self.startButton.setText(_translate("MainWindow",
        ↳ "开始录制"))
173         self.endButton.setText(_translate("MainWindow",
        ↳ "结束录制"))
174         self.shotButton.setText(_translate("MainWindow",
        ↳ "进程中止"))
175         self.menuHelp.setTitle(_translate("MainWindow",
        ↳ "帮助"))
176         self.menu.setTitle(_translate("MainWindow", " 文件"))
177         self.menu_2.setTitle(_translate("MainWindow", "设置"))
178         self.actionOpen_image.setText(_translate("MainWindow",
        ↳ "Open image"))
179         self.actionSave_original_image.setText(_translate("Ma",
        ↳ "inWindow", "Save original
        ↳ image"))
180         self.actionSave_processed_image.setText(_translate("M",
        ↳ "ainWindow", "Save processed
        ↳ image"))
181         self.actionExit.setText(_translate("MainWindow",
        ↳ "Exit"))

```

```
182         self.actionLicense.setText(_translate("MainWindow",
↪         "License"))
183         self.actionAbout.setText(_translate("MainWindow",
↪         "关于"))
184         self.actionmanual.setText(_translate("MainWindow",
↪         "操作说明"))
185         self.action12.setText(_translate("MainWindow",
↪         "设备选择"))
186         self.actionse.setText(_translate("MainWindow",
↪         "文件设置"))
```
