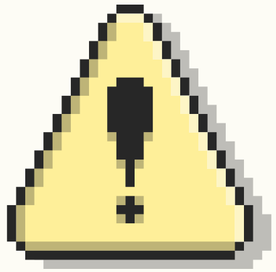
MILK DETECTION ON FOOD LABEL

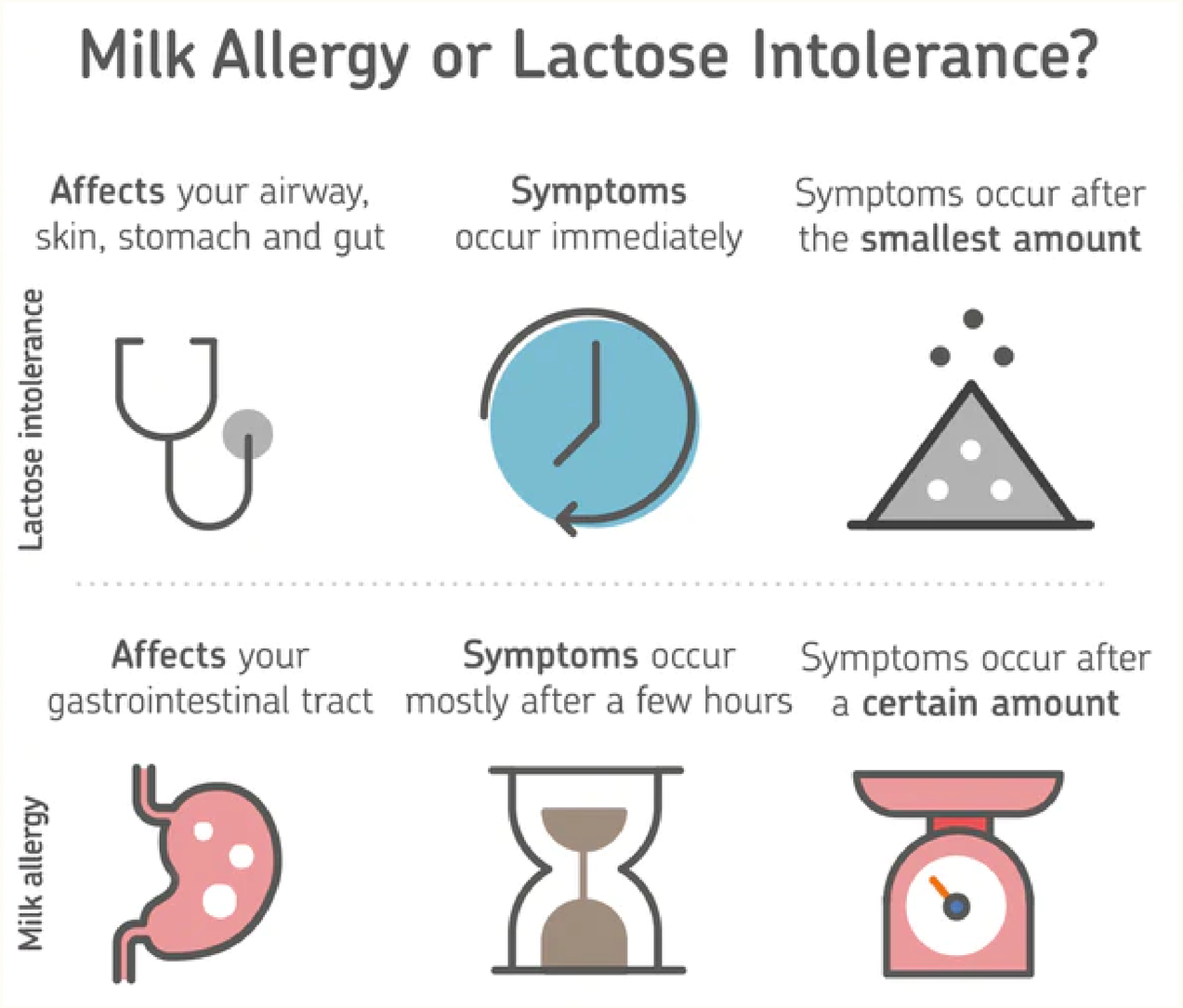
Chawanrat Wisitphongphiboon 6213454 EGBI
Thanyatorn Leethamchayo 6213459 EGBI
Peetimon Arunwiriyahkit 6213462 EGBI

TABLE OF CONTENTS

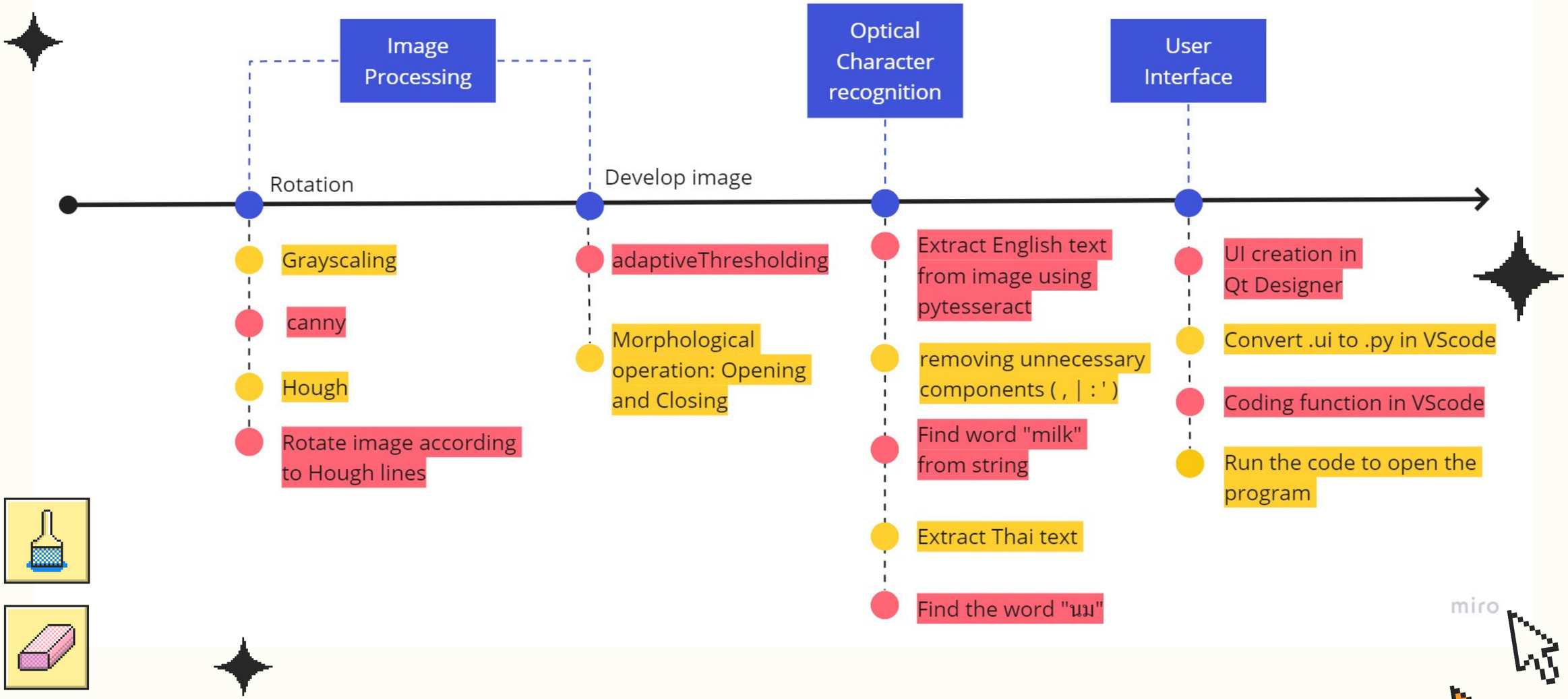
INTRODUCTION



MILK ALLERGY & LACTOSE INTOLERANCE



BLOCK DIAGRAM



CODING STRUCTURE

IMAGE PROCESSING

IMAGE PROCESSING

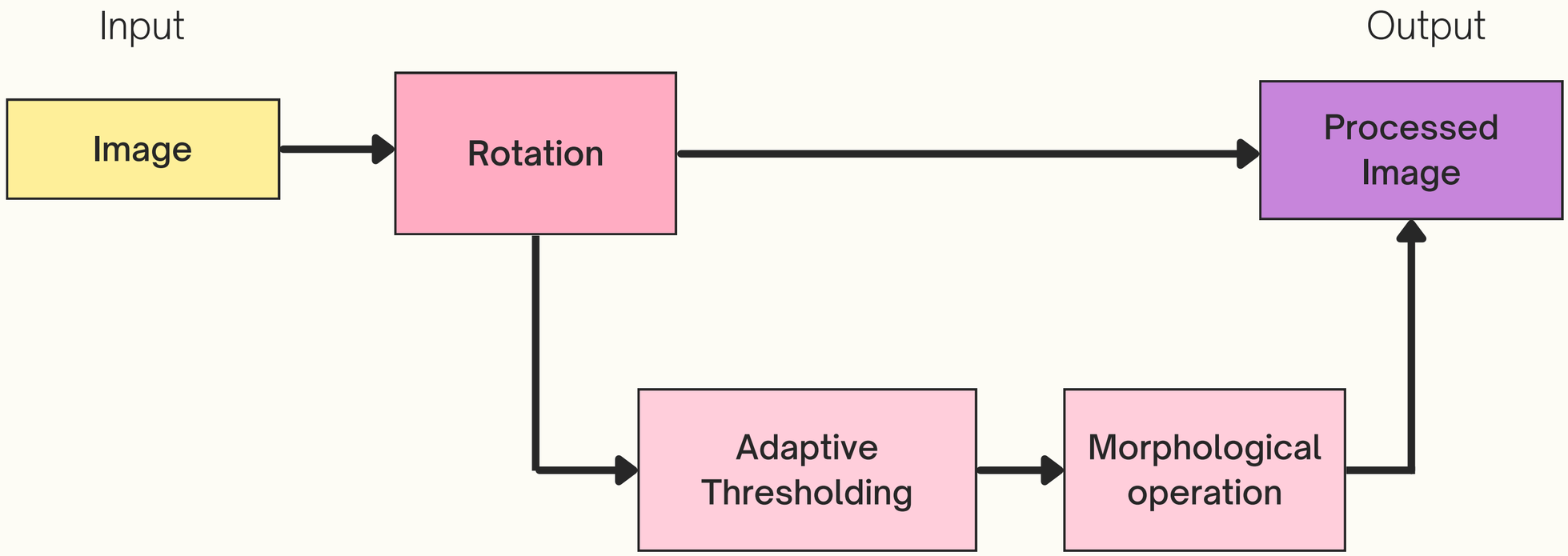
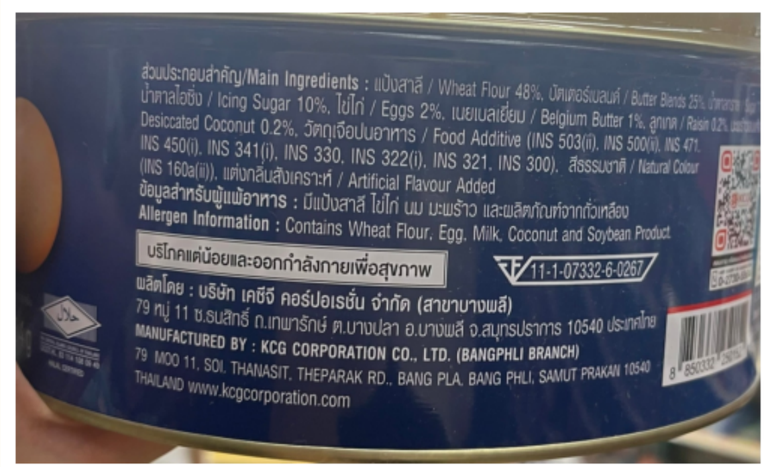


IMAGE ROTATION



1. Grayscaling
2. Canny
3. Hough
4. Rotating

IMAGE ROTATION

1. **Gray scaling**

   - cv2.cvtColor(img, cv2.COLOR_RGB2BGR)
   - cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

2. **Canny**: Edge dectection

   - cv2.Canny(gray_img, 60, 200)

60: lower threshold
200: upper threshold

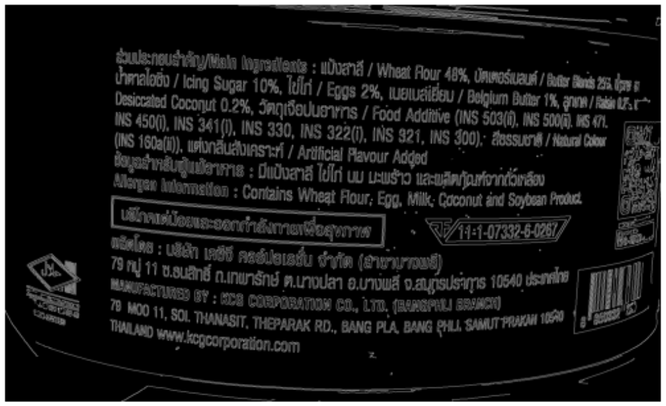
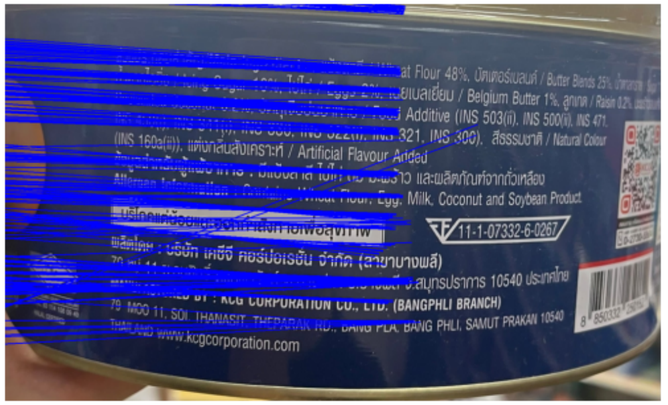


IMAGE ROTATION

3. **Hough**: line detection

- `cv2.HoughLines(canimg, 1, np.pi/180.0, 250, np.array([]))`

  Radian Threshold



4. **Rotate**

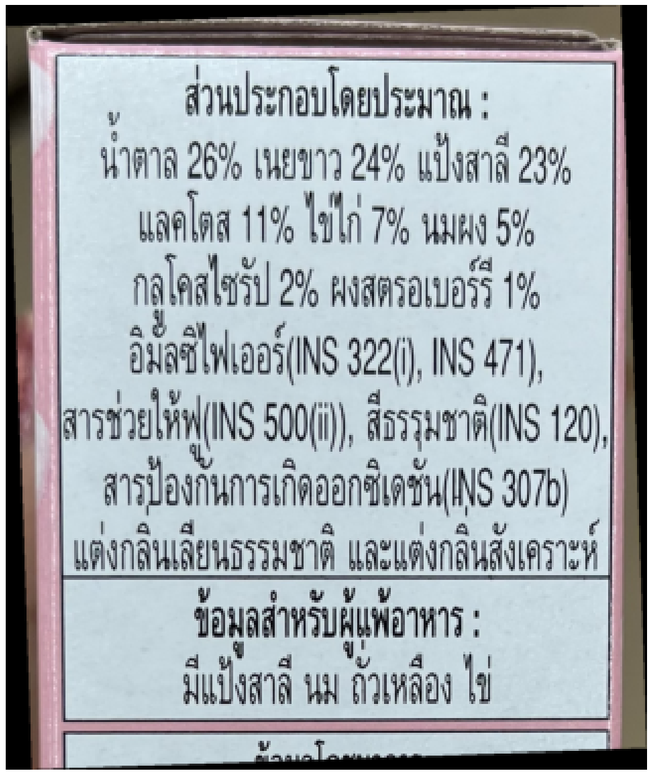
- `180*theta/np.pi - 90`

  Convert back to degree

- `ndimage.rotate(img, mode(deg))`

  Choose mostly found degreeand rotate image

IMAGE PROCESSING



1. **Adaptive thresholding**
2. **Morphological Operation**

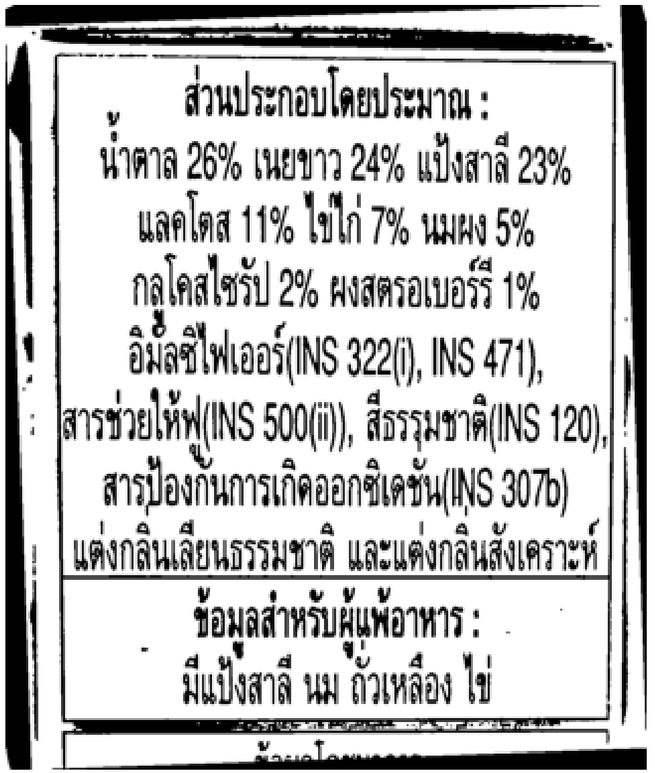


IMAGE PROCESSING

1. **Adaptive thresholding**

- `cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 201, 9)`

Threshold

Binary thresholding

gaussian-weighted threshold value

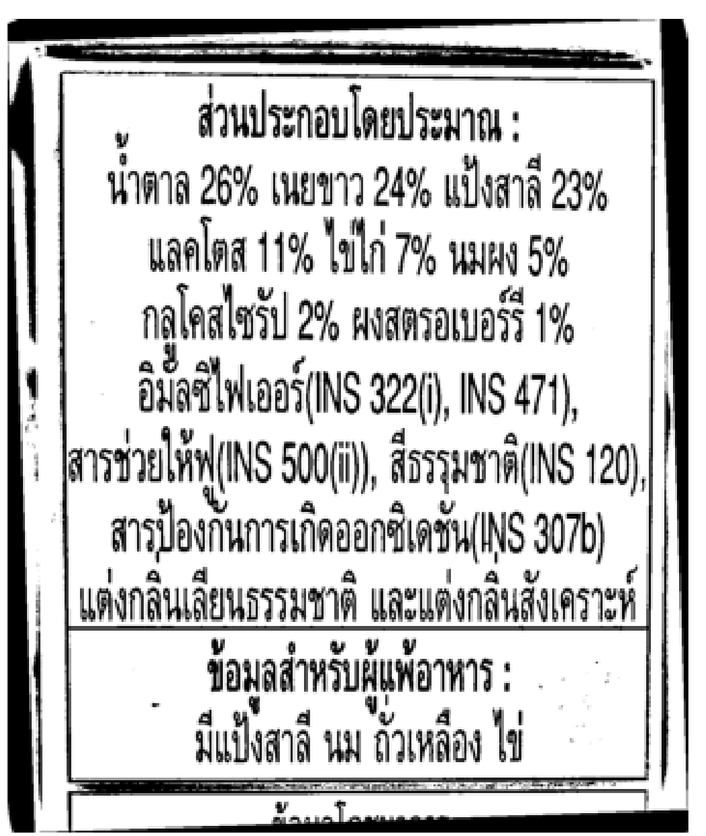


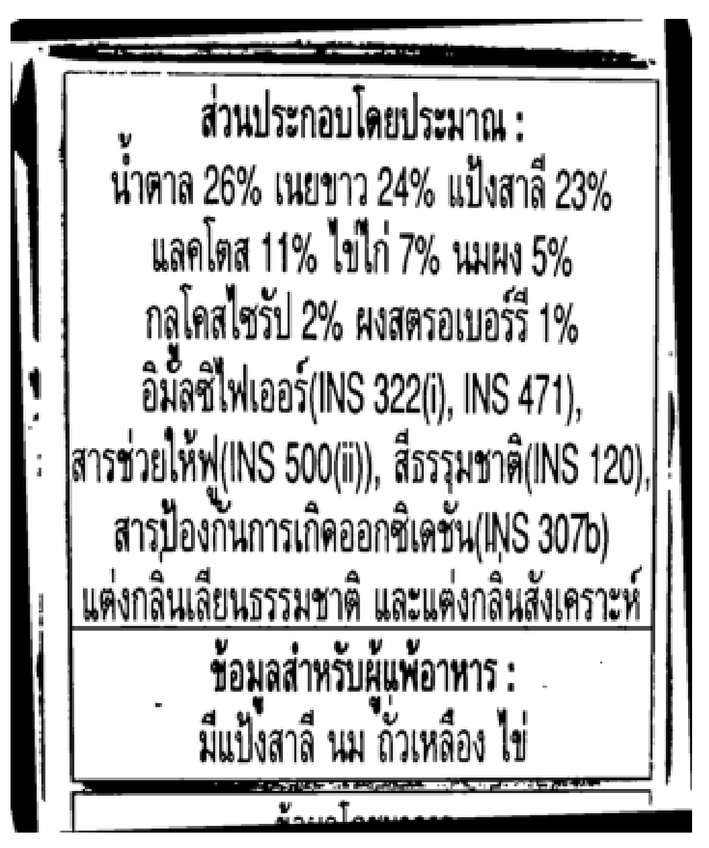
IMAGE PROCESSING

2. **Morphology Operation**

- Get kernel
  ```
  cv2.getStructuringElement
  (cv2.MORPH_RECT,(3,3))
  ```

- Opening
  ```
  cv2.morphologyEx(adapthresh,
  cv2.MORPH_OPEN, kernel)
  ```

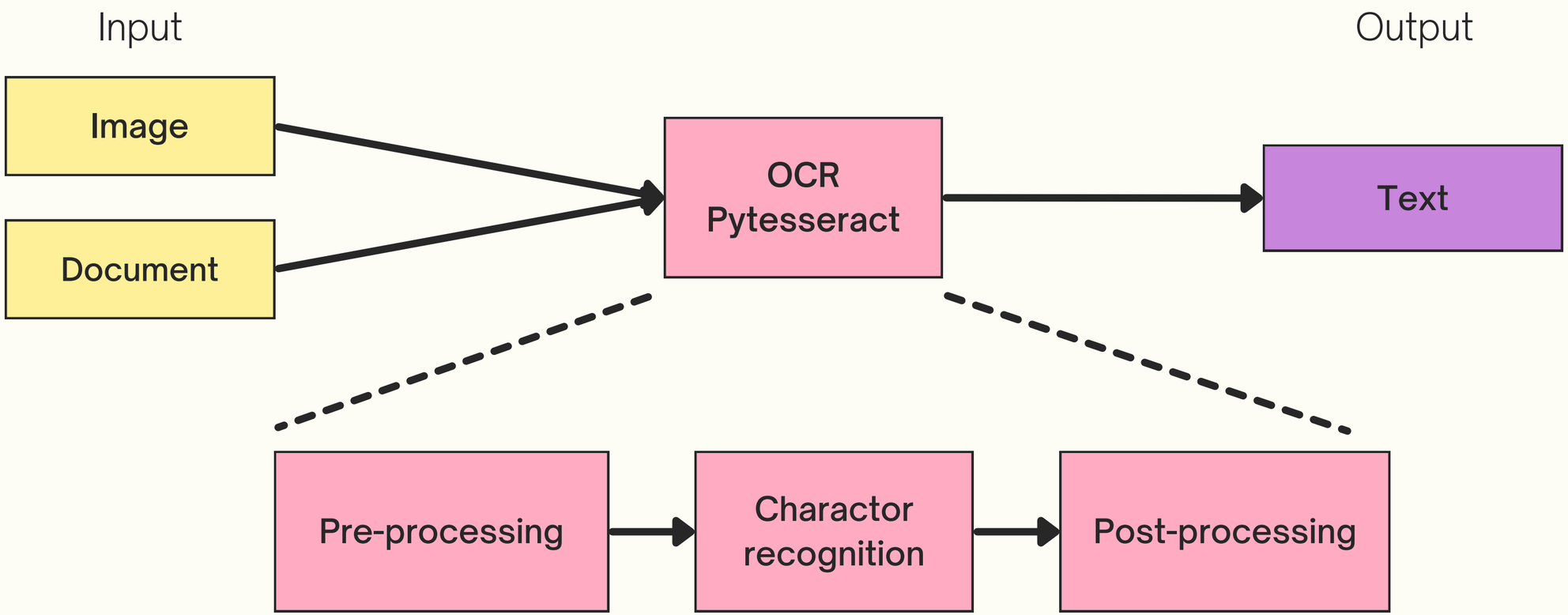
- Closing
  ```
  cv2.morphologyEx(opening,
  cv2.MORPH_CLOSE, kernel)
  ```



CODING STRUCTURE

OCR

OPTICAL CHARACTER RECOGNITION (OCR)



OCR - PYTESSERACT

1. Extract text

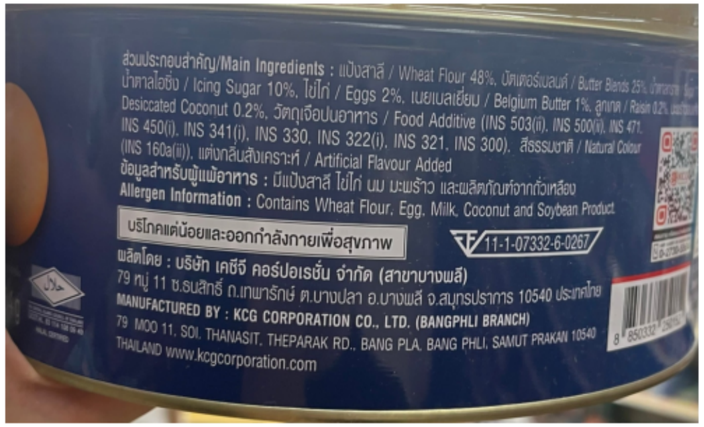
English

```
data = pytesseract.image_to_data(img,
output_type=pytesseract.Output.DICT)
```

Thai

```
data_th = pytesseract.image_to_data(img,
output_type=pytesseract.Output.DICT,
lang="tha")
```

data and data_th are dictionary variables that contain many keys




```
dous:neudrAny/Main Ingredients : l0ve7a / Wheat Flour 48%, Uninasivau / Butter
Blends 25% Umass Ge
thmaleiu / Icing Sugar 10%, Tuln / Eggs 2%, 1uslUaleU / Belgium Butter 1%, gina /
Rash 02%
Desiccated Coconut 0.2%, Sanisavuarms / Food Additive (INS 503(i), INS 500(i, INS
471
INS 450(i), INS 341(i), INS 330, INS 322(j), INS 321, INS 300), aSssuuh / Natura
Coou
(INS 160a(i), urlonauavinsr:r / Artificial Flavour Added ==
dayadnsuduiowms ; duOvend Twin wy u-wsr9 Wa:wannrurionnoinaad

Allergen Information : Contains Wheat Flour, Egg, Milk, Coconut and Soybean Pott

TEETEOEOD

wanlny : U8En N83 MasUaIsGu daria (aww DUA) s4o us
79 ni) 11 vsuand n.nurgnd murovan e.uwwa o.aynsusims 10
NIANUFACTURED BY : KCG CORPORATION CO., LTD. (BANGPHL BRANCH!

4 M00 11, SOI. THANASIT, THEPARAK RD., BANG PLA, BANG PHLI, SAMUT P
AILAND Wwww.kegcorporation.com

RAKAN 40540
```

OCR - PYTESSERACT

2. Recognize word

- Delete unnecessary string (,|:;')

```
for i, word in enumerate(data["text"]):
  data["text"][i] = re.sub(',|:', '', word)
```

- English

```
wordrecog_en = [ i for i, text in enumerate(data["text"]) if text.lower() == 'milk']
```

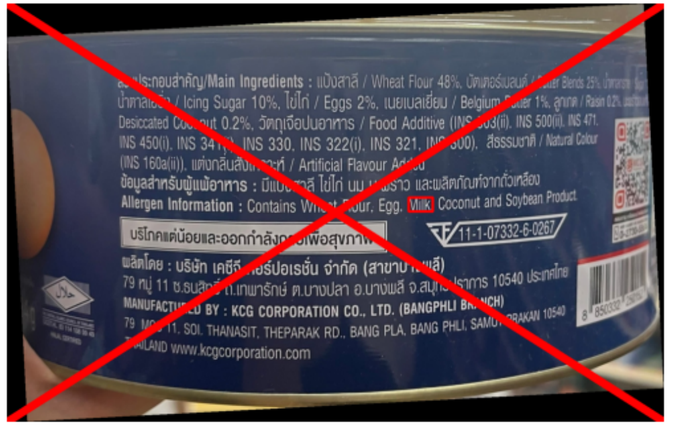
- Thai

```
wordrecog_m = [ i for i, text in enumerate(data_th['text']) if text == 'ม']
wordrecog_k = [ i for i, text in enumerate(data_th['text']) if text == 'น']
wordrecog_th = []
wordrecog_th = [ i for i, text in enumerate(data_th['text']) if text == 'มน']
```

OCR - PYTESSERACT

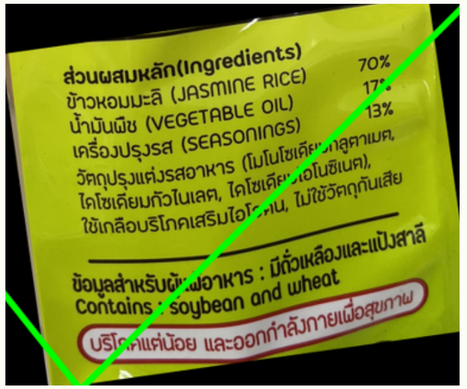
3. Visualization

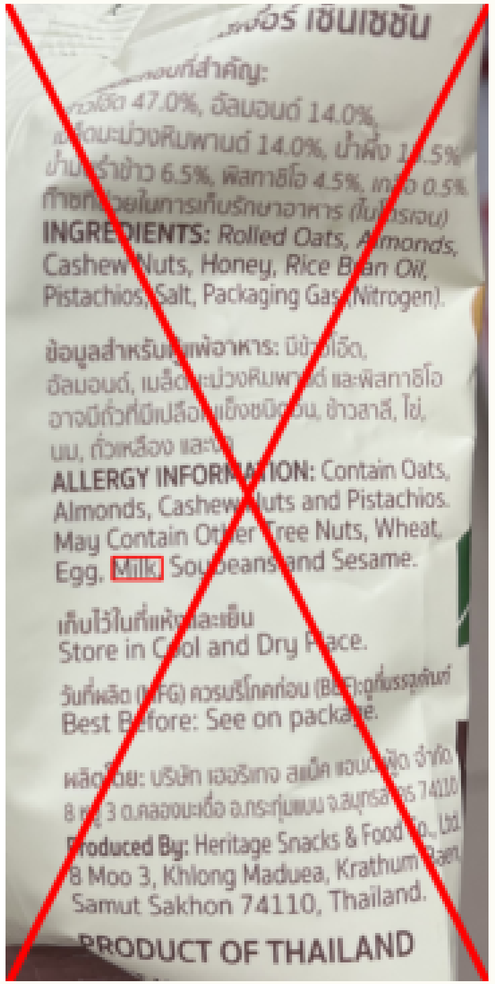
- **Draw a box** around the word 'milk' or 'uu' by using coordinates from keys of the dictionary variable: width, height, left, and top
- **Draw x** on top of an image that contains milk
- **Draw ✓** on top of an image that does not contain milk

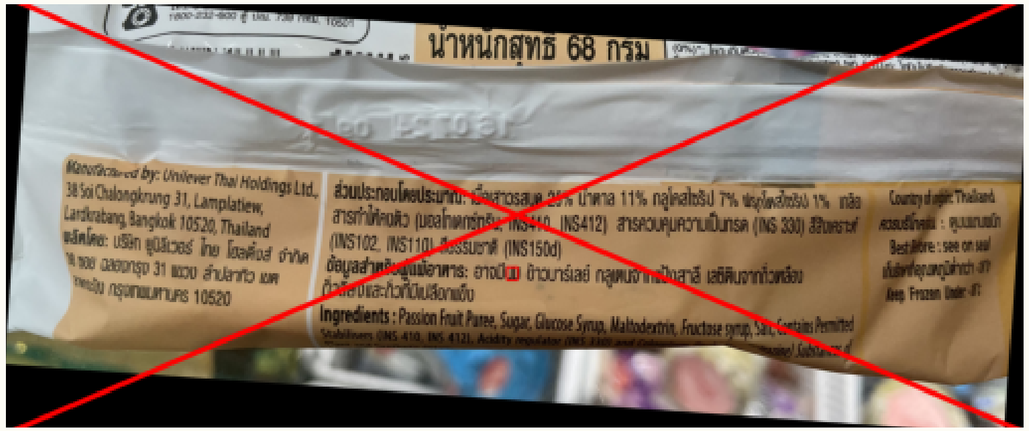


OCR - PYTESSERACT

Example results



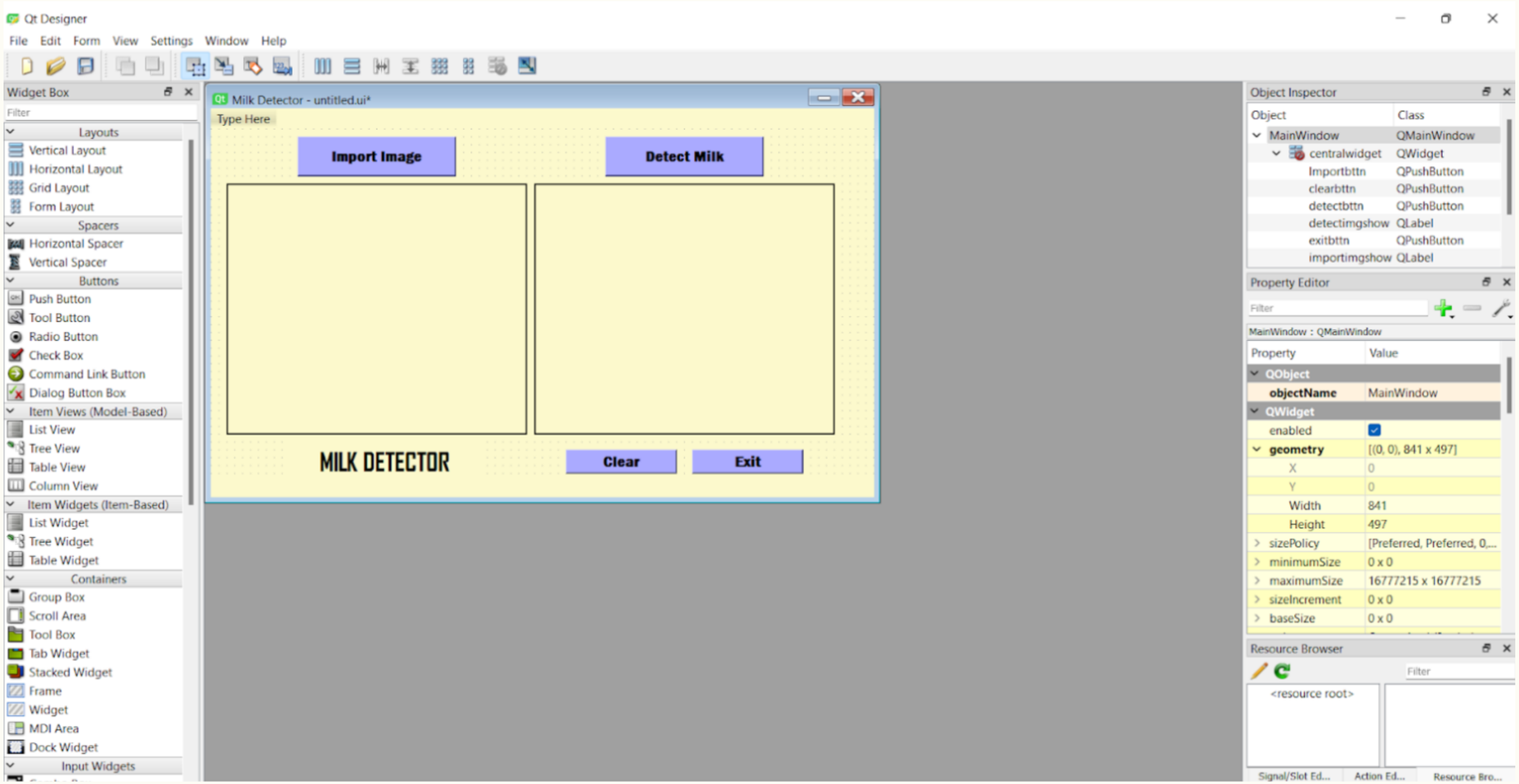




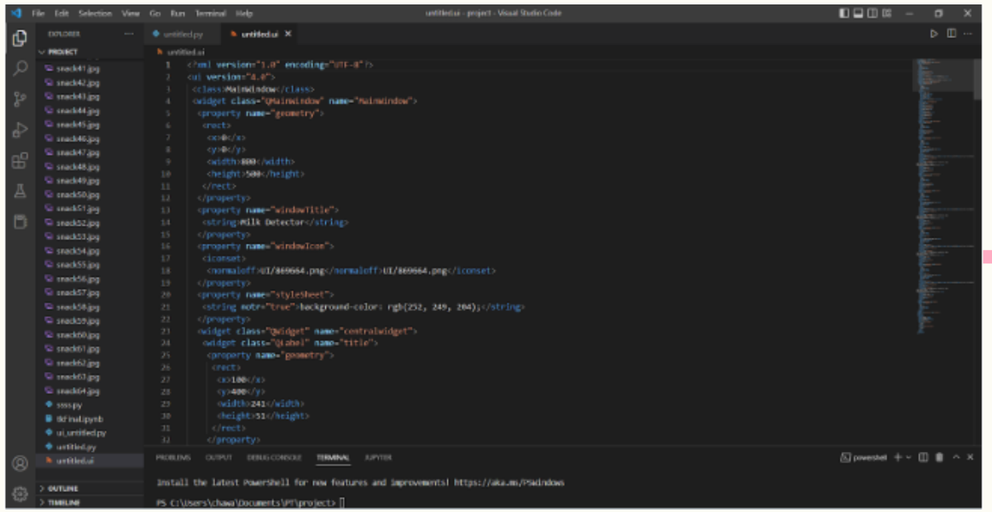
CODING STRUCTURE

UI - MILK DETERTOR

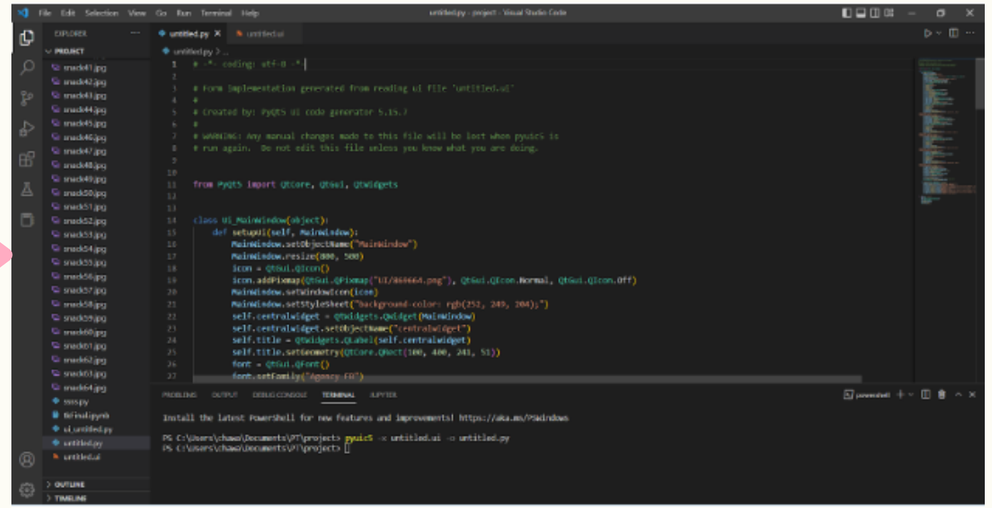
UI CREATION: QT DESIGNER



UI CREATION: FILE CONVERSION



pyuic5



UI CREATION: CODING

```
        def browseImage(self):
                options = QtWidgets.QFileDialog.Options()
                options |= QtWidgets.QFileDialog.DontUseNativeDialog
                self.fileName, _ =
QtWidgets.QFileDialog.getOpenFileName(options=options)

                if self.fileName:
                        pattern = ".(jpg|png|jpeg|bmp|jpe|tiff)$"
                        if re.search(pattern,self.fileName):
                                self.setImage(self.fileName)

        def putImageinspace(self,fileName):
                self.importimgshow.setPixmap(QPixmap(fileName))
                self.detectbttn.setEnabled(True)
```

```
                #if clicked "Import Image" button, the function "getImage" will
work and image will be imported
                self.Importbttn.clicked.connect(self.browseImage)
```

Import Image

```
        def detectmilk(self):
                # read non-rotated img
                img_nr = cv2.imread(self.fileName)
                img_nr = cv2.cvtColor(img_nr, cv2.COLOR_RGB2BGR)

                # ---------- Rotate image ---------------

                # change image to grayscale
                gray_ori = cv2.cvtColor(img_nr, cv2.COLOR_BGR2GRAY)
                # Apply canny
                canimg = cv2.Canny(gray_ori, 60, 200)
                # Apply Hough
                lines = cv2.HoughLines(canimg, 1, np.pi/180.0, 250, np.array([]))
```

```
                #if clicked "Detect Milk" button, the function "detectmilk" will work
nd detected image will shown
                self.detectbttn.clicked.connect(self.detectmilk)
```

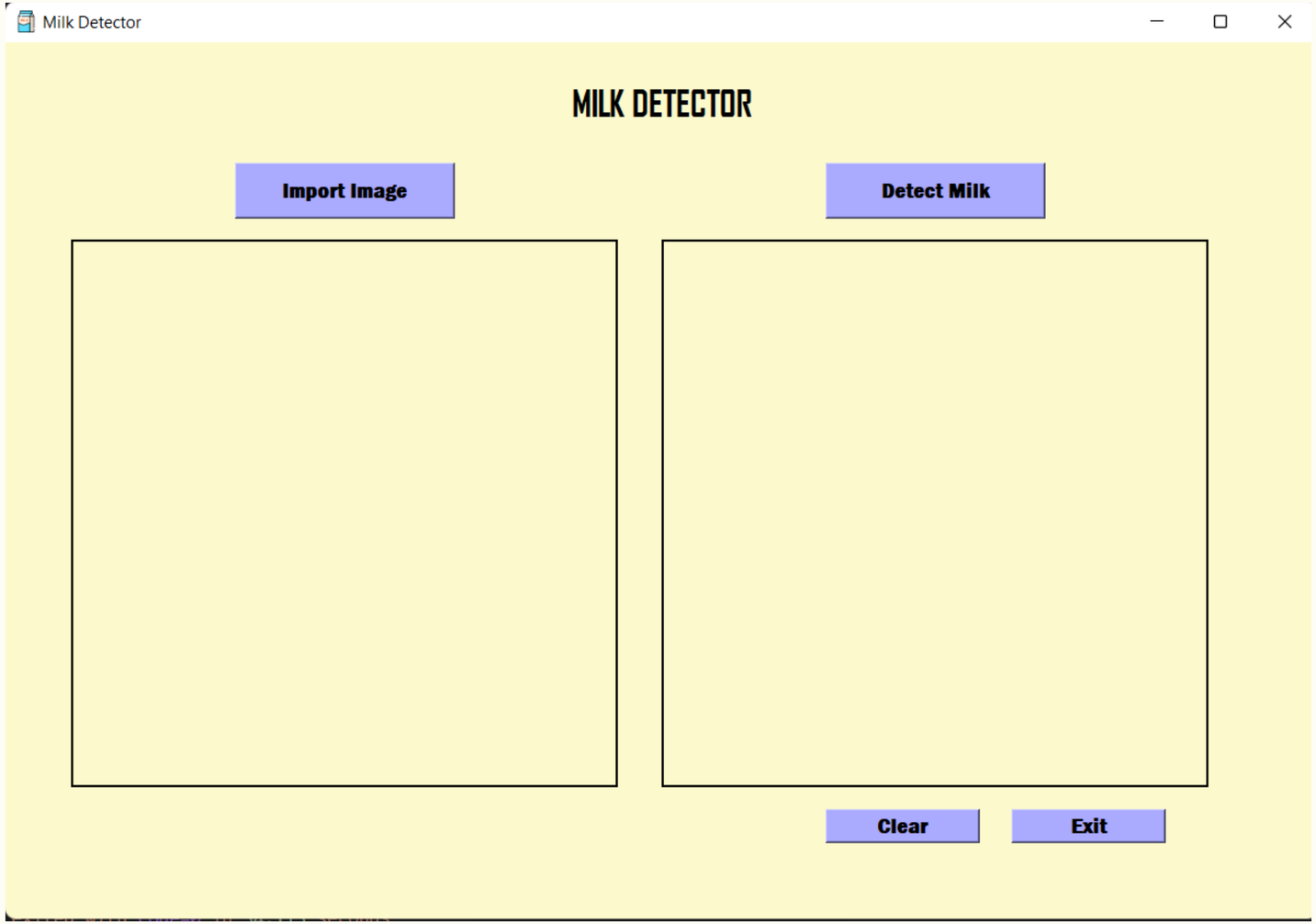
Detect Milk

```
    def clearimage(self):
            self.detectimgshow.clear()
```

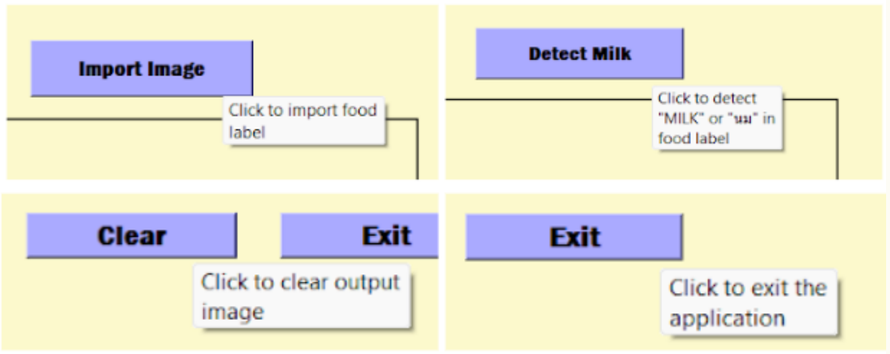
```
    #if click "Clear", the detected image will be clear out
    self.clearbttn.clicked.connect(self.clearimage)
```

Clear

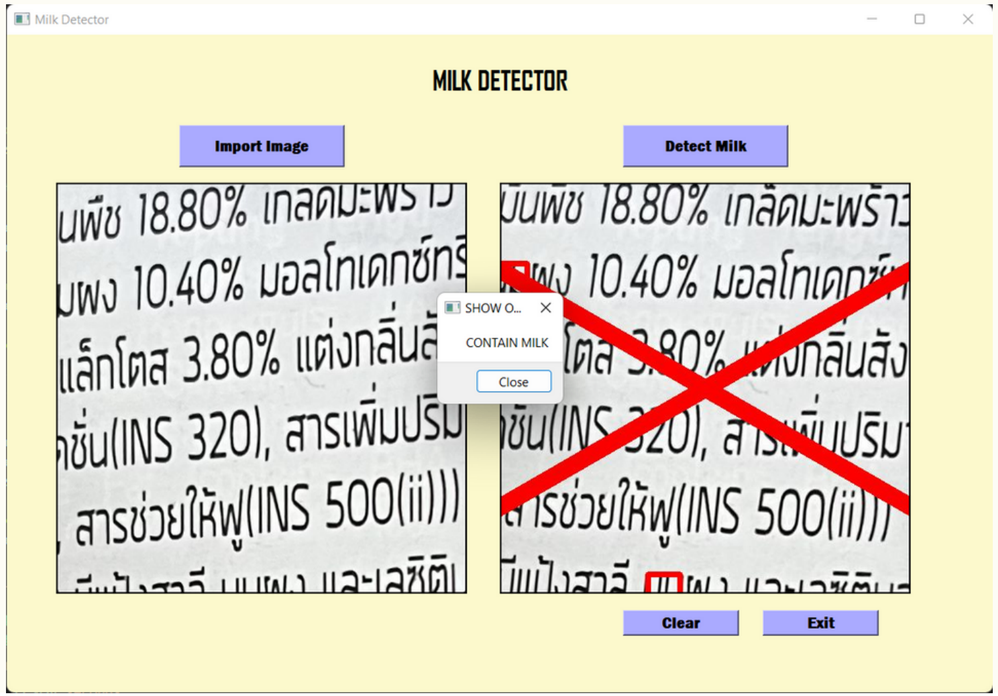
MILK DETECTOR

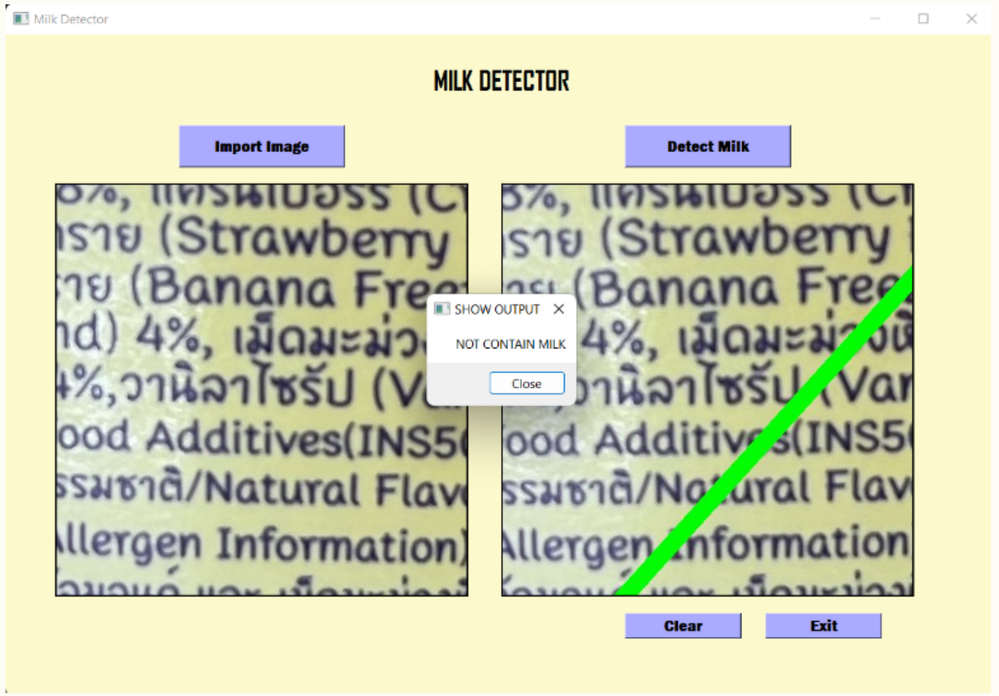


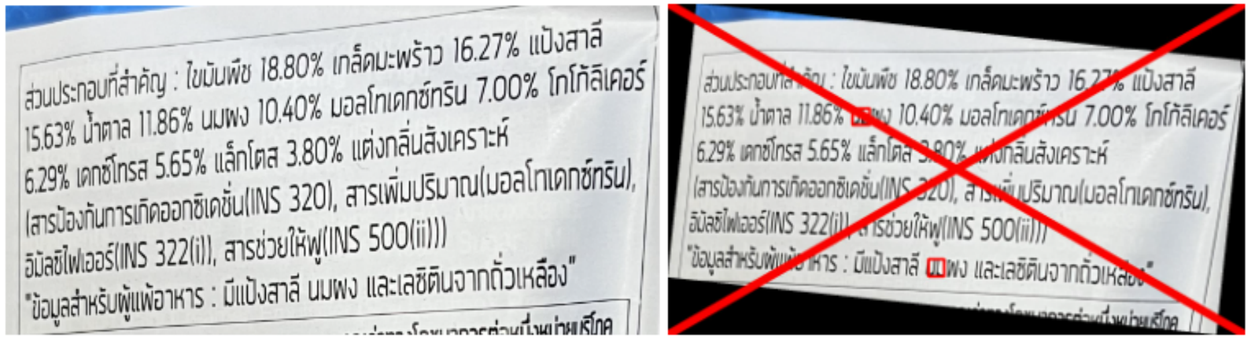
Tooltips

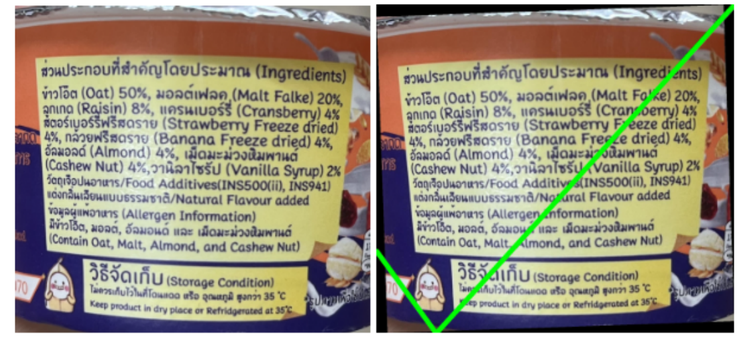


RESULT


Milk Detector
MILK DETECTOR
Import Image
Detect Milk
SHOW O...
CONTAIN MILK
Close
Clear
Exit


Milk Detector
MILK DETECTOR
Import Image
Detect Milk
SHOW OUTPUT
NOT CONTAIN MILK
Close
Clear
Exit


ส่วนประกอบที่สำคัญ : ไขมันพืช 18.80% เกล็ดมะพร้าว 16.27% แป้งสาลี
15.63% น้ำตาล 11.86% นมผง 10.40% มอลโทเดกซ์ทริน 7.00% โกโก้ลิเคอร์
6.29% เดกซ์โทรส 5.65% แล็กโตส 3.80% แต่งกลิ่นสังเคราะห์
(สารป้องกันการเกิดออกซิเดชั่น(INS 320), สารเพิ่มปริมาณ(มอลโทเดกซ์ทริน),
อิมัลซิไฟเออร์(INS 322(i)), สารช่วยให้ฟู(INS 500(ii)))
"ข้อมูลสำหรับผู้แพ้อาหาร : มีแป้งสาลี นมผง และเลซิตินจากถั่วเหลือง"

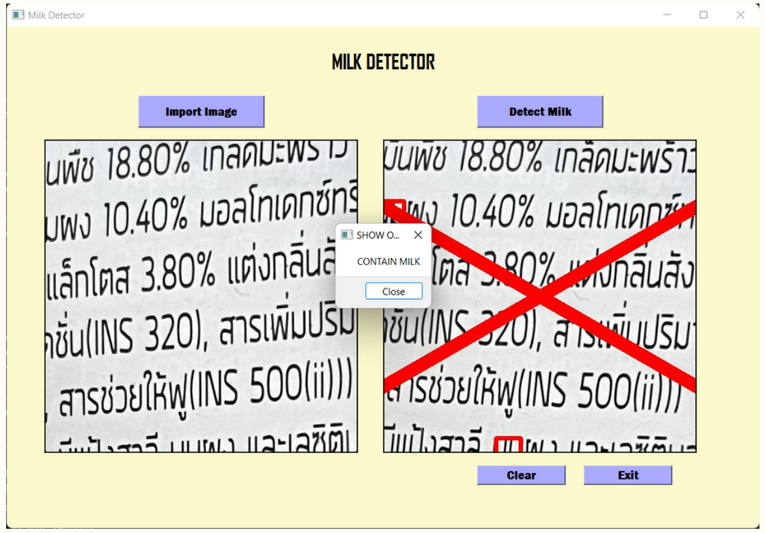

ส่วนประกอบที่สำคัญโดยประมาณ (Ingredients)
ข้าวโอ๊ต (Oat) 50%, มอลต์เฟลค (Malt Falke) 20%,
ลูกเกด (Raisin) 8%, แครนเบอร์รี่ (Cransberry) 4%
สตอร์เบอร์รีฟรีซดราย (Strawberry Freeze dried)
4%, กล้วยฟรีซดราย (Banana Freeze dried) 4%,
อัลมอลด์ (Almond) 4%, เม็ดมะม่วงหิมพานต์
(Cashew Nut) 4%,วานิลาไซรัป (Vanilla Syrup) 2%
วัตถุเจือปนอาหาร/Food Additives(INS500(ii), INS941)
แต่งกลิ่นเลียนแบบธรรมชาติ/Natural Flavour added
ข้อมูลผู้แพ้อาหาร (Allergen Information)
มีข้าวโอ๊ต, มอลต์, อัลมอนด์ และ เม็ดมะม่วงหิมพานต์
(Contain Oat, Malt, Almond, and Cashew Nut)
วิธีจัดเก็บ (Storage Condition)
Keep product in dry place or Refridgerated at 35°C.

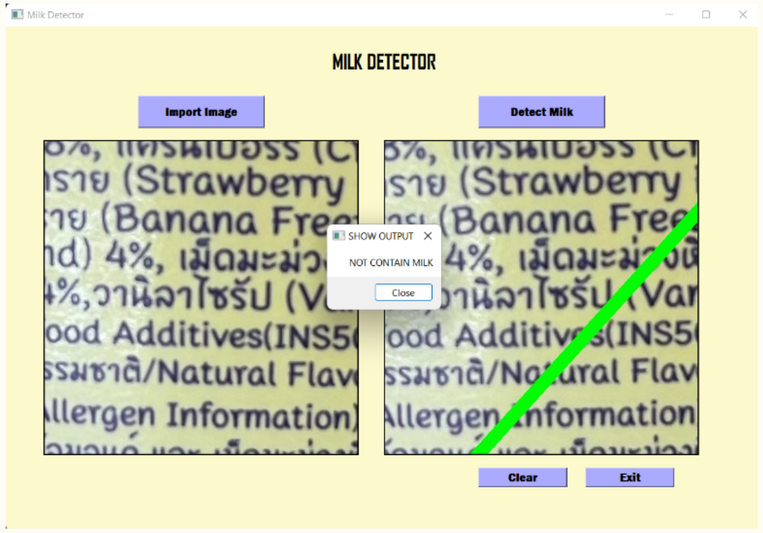
RESULT

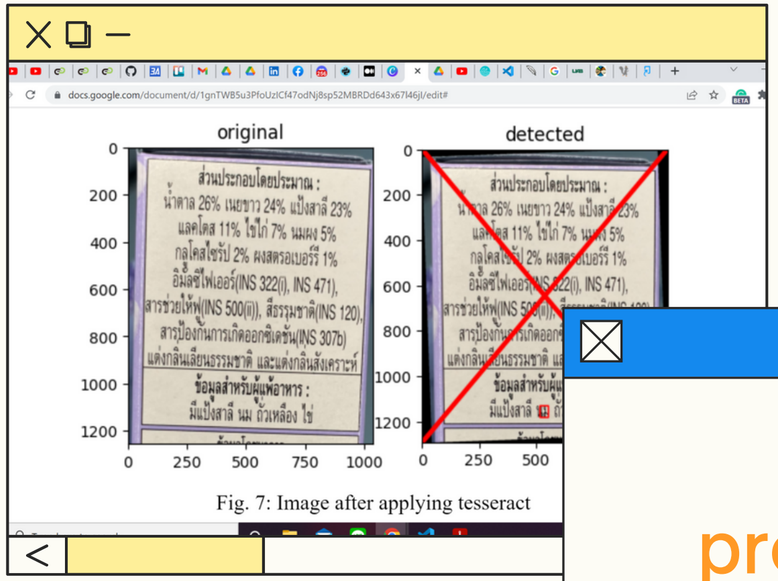
73.26% accuracy

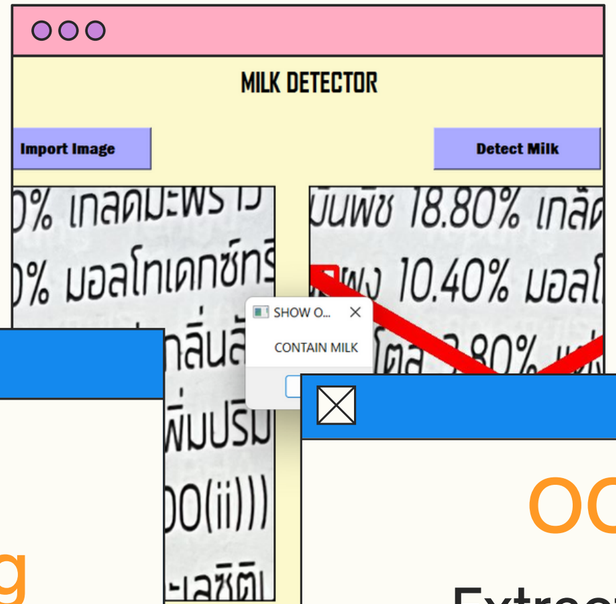
A total of 86 images:

- 63 predict correctly
- 23 predicted incorrectly
  - 1 False Positive (predicted non-dairy products as containing milk)
  - 22 False Negatives (predicted dairy products as not containing milk)









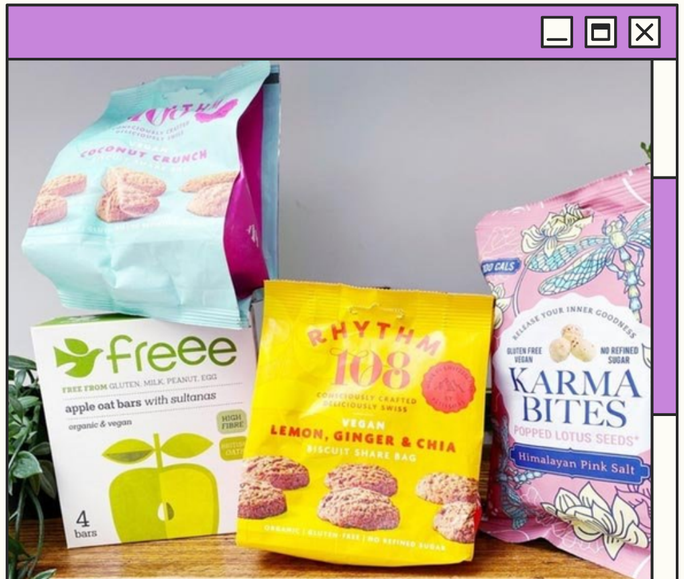
SUMMARY

Image processing

Rotating image and noise filtering

OCR

Extract text, Recognize word, and Visualization



UI

UI creation by Qt Designer, File conversion, and coding

FUTURE IMPROVEMENT

- Add more parameters for food allergic apart from milk
- Using Barcode to scan for check food allergy
- Better database and data acquisition system to protect personal information
- More user-friendly UI design
- Apply machine learning to improve accuracy

THANK YOU!