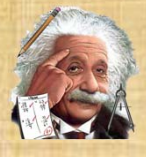
Design Thinking By AI

Surawich Aungsusink 6310422083



1 | Empathize

ทำความ
เข้าใจ

Interview

Empathize

Noted from your interview

Name: Autwachi

Profile: Thai Traditional Doctor

Life Style: Trekking, Colleting Herbs and Compouding Medecine

2

Define

ตีโจทย์

กำหนดกรอบของปัญหา

DEFINE

User Description Autwachi is a Thai Traditional Doctor who working in almost very day in Chanthaburi.

Need a way to (user's need) Autwachi needs to verify plant morphology before collecting herbs and compounding medicine for patient.

Surprisingly/ Because/But (user's insight)

It is normally difficult to visual check a different type of herb leaves as herb must be always collected in the morning time. He rarely meets the target because of time constraint.

3 Ideate

ระดมความคิด

Sketch solution to meet your user's needs

IDEATE

- Plant herbs in the garden nearby his home.
- Hire more people to collect herb and train them to check herb leaves from photograph.
- Use drone to find out most possible area for collecting herbs.
- Use google map for faster trekking.
- Use robot to help collecting herbs.
- Use AI and image processing to identify plant morphology.
- … etc

4

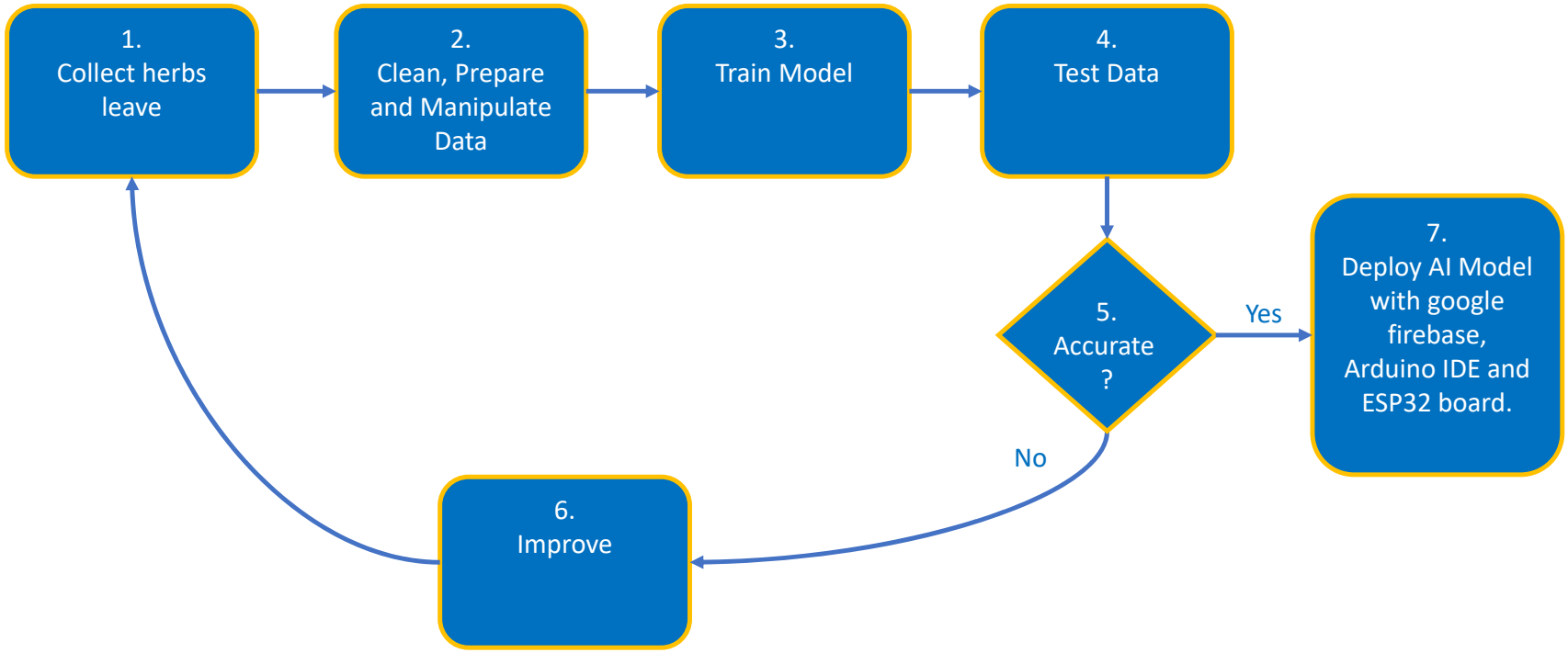
prototype

สร้าง
ต้นแบบ

Draw your prototype

PROTOTYPE

- Use AI and image processing to identify plant morphology.



5 test

ทดสอบ

Share your solution and get feedback test

Like

- Feels more comfortable to have supporter.
- Feels happy to save time for identify herb leaves.

Dislike

- Feels too expensive to invest in new technology.
- Does not want to trust 100 percent in AI.

Question

- How can to take picture for all herb leaves ?
- How can AI detect the right herb leaves ?
- Is there any ways to buy a cheaper camera and a cloud database ?
- How can real time connect ESP232 / Arduino IDE board (IOT equipment) with cloud database to detect herb leaves image ?

Idea

- Gradually download the target herb leaves from internet as initial train data.
- The more training data and testing data with good technique such as CNN, ConvNet or GoogLeNet can enhance the better detection.
- Use 12 month-free trial cloud database at the initial state such as Huawei Cloud.
- Use LORA WAN to connect IOT equipment with cloud database.

Appendix I – app.py

```python
import cv2
import numpy as np
import dlib
import pickle

detector = dlib.get_frontal_face_detector()
sp = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
facerec = dlib.face_recognition_model_v1('dlib_face_recognition_resnet_model_v1.dat')
FACE_DESC,FACE_NAME = pickle.load(open('trainset.pk','rb'))

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
font = cv2.FONT_HERSHEY_COMPLEX
cam = cv2.VideoCapture(0)

buffer = (30,10)
while True:
    _, img = cam.read()
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)
    for (x,y,w,h) in faces:
        r1 = y - buffer[0] if y-buffer[0] >= 0 else 0
        r2 = y + h + buffer[0] if y + h + buffer[0] < img.shape[0] else img.shape[0]-1
        c1 = x - buffer[1] if x-buffer[1] >= 0 else 0
        c2 = x + w + buffer[1] if x + w + buffer[1] < img.shape[1] else img.shape[1]-1
        face = gray[r1:r2,c1:c2]

        dets = detector(img, 1)
        name = 'unknown'
        dmin = 0
        for k, d in enumerate(dets):
            shape = sp(img, d)
            face_descriptor = np.array(facerec.compute_face_descriptor(img, shape))

            d = []
            for face_desc in FACE_DESC:
                d.append(np.linalg.norm(face_descriptor - face_desc))
            d = np.array(d)
            idx = np.argmin(d)
            if d[idx] < 0.5:
                name = FACE_NAME[idx]
                dmin = d[idx]
                print(name, dmin)
        cv2.putText(img, name+':{:.2f}'.format(dmin), (x,y-5), font, .7, (255,0,0), 1)
        cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)


    cv2.imshow('img', img)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

Appendix II – trainpicture.py

```python
import numpy as np
import cv2
import dlib
import os
import pickle

path = './facedata/'
detector = dlib.get_frontal_face_detector()
sp = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
facerec = dlib.face_recognition_model_v1('dlib_face_recognition_resnet_model_v1.dat')

FACE_DESC=[]
FACE_NAME=[]
for fn in os.listdir(path):
    if fn.endswith('.jpg'):
        img = cv2.imread(path+fn)
        dets = detector(img, 1)
        for k, d in enumerate(dets):
            shape = sp(img, d)
            face_descriptor = facerec.compute_face_descriptor(img, shape)
            FACE_DESC.append(np.array(face_descriptor))
            print('loading...',fn)
            FACE_NAME.append(fn[:fn.index('_')])

pickle.dump((FACE_DESC,FACE_NAME),open('trainset.pk','wb'))
```