RNF SYSTEM
READ AND FETCH

Created by Suruj Kalita

Read and Fetch is a system which allows you to collect data from user and store it in the form of photos , it can be applicable on both hardware and software side .

Reading the data

Processing the data

Storing it in the Folder

Abilities

- It can give you minute details capture from the camera
- Applicable on any system with or without inbuilt camera.

What special about this System is that , it is tunable according to user , easy to modify , easy to use .

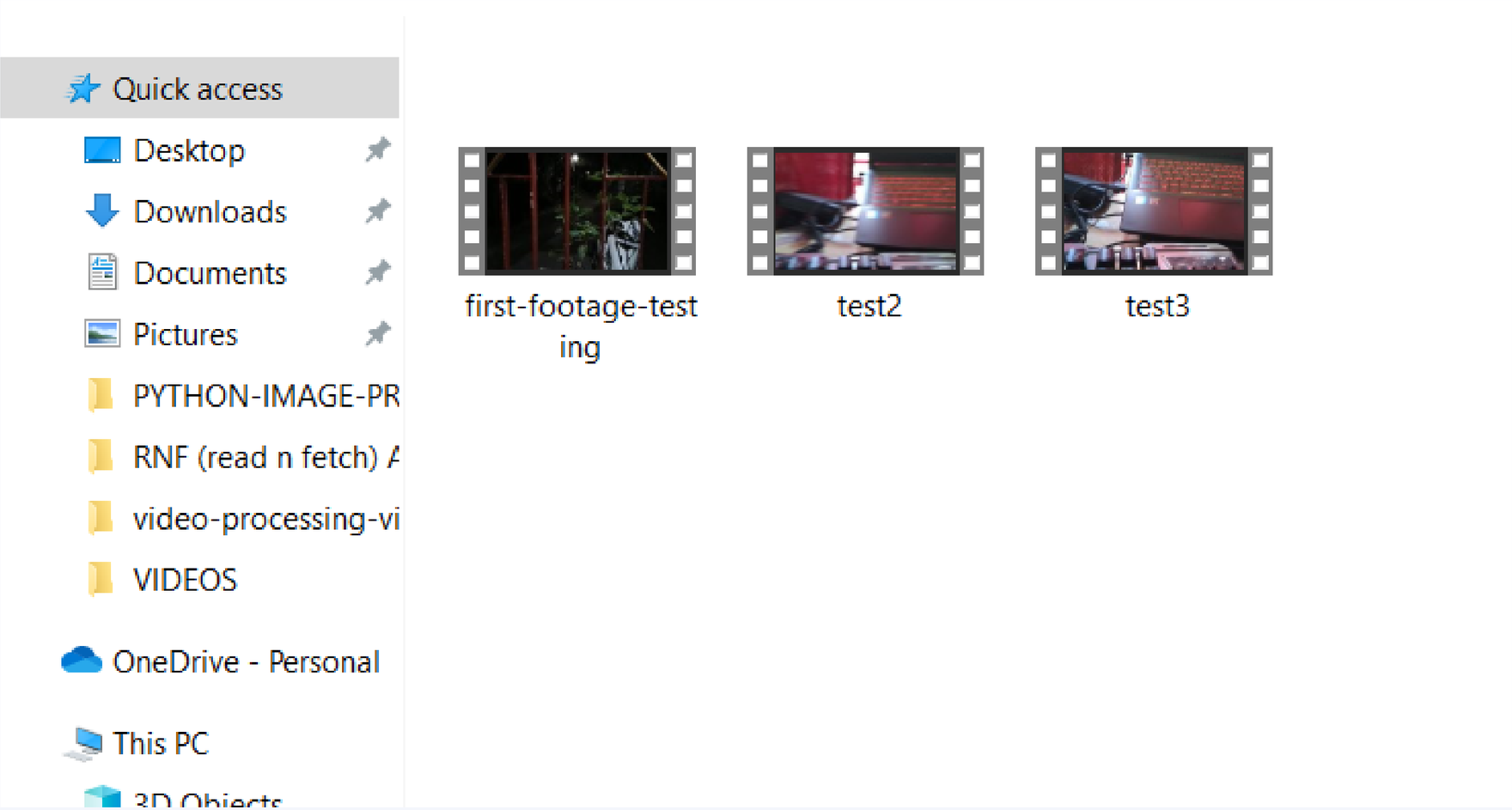
Prototype of the compilers : --

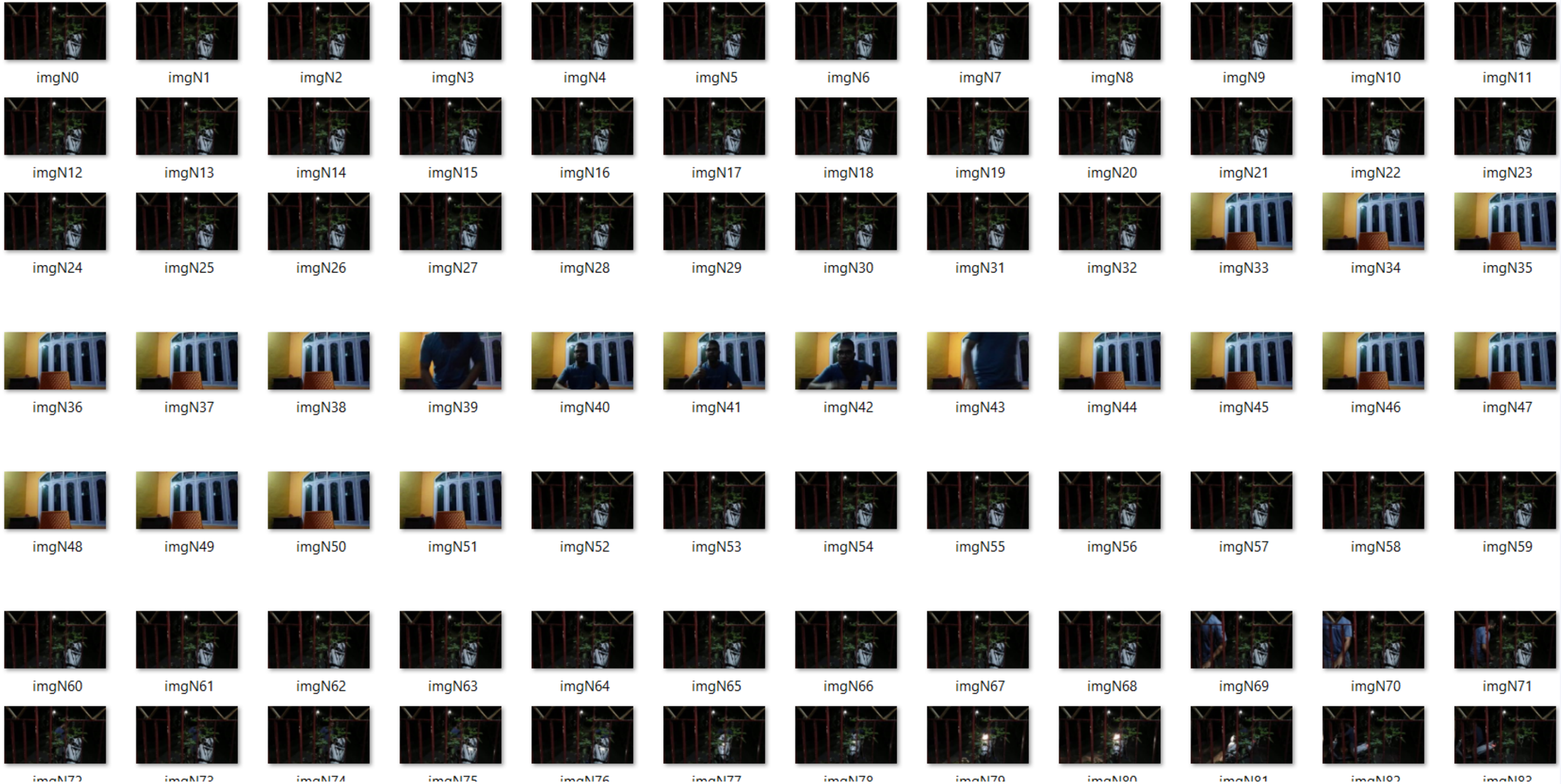
First compiler to access additional camera if the system lacks inbuilt camera .

```
while(cap.isOpened()):
    ret , frame = cap.read()
    if ret == True:
        #frame = cv2.resize(frame , (1920 , 1080)) # for camera use frame size of high resolution (1920 , 1
        cv2.imshow("color-frame", frame)
        output.write(frame)
        if cv2.waitKey(1) == ord("q"):
            break

cap.release()
output.release()
cv2.destroyAllWindows()
```

Second Compiler : -- To process the data , that is fetch from the additional camera . What special about this compiler is that it is tunable, if the system user is using already contain inbuilt camera than the user can directly retrieve data from this complier and store the data in target folder set by the user

```
vidcap = cv2.VideoCapture()  # put o if you want to access system camera for reading the videos
ret,image = vidcap.read() #read the video

count = 0 # COUNTER
while True:
    if ret == True:
        #cv2.imwrite(r" ---- \imgN%d.jpg"%count,image)
        cv2.imwrite(r"C:\Users\suruj\PYTHON-IMAGE-PROCESSING\RNF (read n fetch) API\FRAME1\imgN%d.jpg"%count,image)
        vidcap.set(cv2.CAP_PROP_POS_MSEC,(count**100)) #SETTING THE SPEED
        ret,image = vidcap.read()
        cv2.imshow("frame",image)
        #print(count)
        count +=1
        if cv2.waitKey(1) == ord("q"):
            break
            cv2.destroyAllWindows()

vidcap.release()
cv2.destroyAllWindows()
```


Quick access
Desktop
Downloads
Documents
Pictures
PYTHON-IMAGE-PR
RNF (read n fetch) A
video-processing-vi
VIDEOS
OneDrive - Personal
This PC
3D Objects
first-footage-testing
test2
test3


imgN0
imgN1
imgN2
imgN3
imgN4
imgN5
imgN6
imgN7
imgN8
imgN9
imgN10
imgN11
imgN12
imgN13
imgN14
imgN15
imgN16
imgN17
imgN18
imgN19
imgN20
imgN21
imgN22
imgN23
imgN24
imgN25
imgN26
imgN27
imgN28
imgN29
imgN30
imgN31
imgN32
imgN33
imgN34
imgN35
imgN36
imgN37
imgN38
imgN39
imgN40
imgN41
imgN42
imgN43
imgN44
imgN45
imgN46
imgN47
imgN48
imgN49
imgN50
imgN51
imgN52
imgN53
imgN54
imgN55
imgN56
imgN57
imgN58
imgN59
imgN60
imgN61
imgN62
imgN63
imgN64
imgN65
imgN66
imgN67
imgN68
imgN69
imgN70
imgN71

```python
import cv2
```

Python

```python
camera = "http://192.168.25.84:8080//video"  # /video is neccesory
```

Python

```python
cap = cv2.VideoCapture(0)
```

Python

```python
cap.open(camera)
```

Python

```
True
```

```python
print("check===" , cap.isOpened())
```

Python

```
check=== True
```

```python
fourcc = cv2.VideoWriter_fourcc(*"XVID")
```

Python

Design to access additional camera and further saving it to your system

SATCHEL PAIGE

```python
while(cap.isOpened()):
    ret , frame = cap.read()
    if ret == True:
        #frame = cv2.resize(frame , (1920 , 1
        cv2.imshow("color-frame", frame)
        output.write(frame)
        if cv2.waitKey(1) == ord("q"):
            break

cap.release()
output.release()
cv2.destroyAllWindows()
```

Design of the fetching algorithm of the system It can be directly used by accessing the front camera of your system or can be used to fetch the data capture from the additional camera .

```
vidcap = cv2.VideoCapture()  # put o if you want to access system camera for reading the videos
ret,image = vidcap.read() #read the video

count = 0 # COUNTER
while True:
    if ret == True:
        #cv2.imwrite(r" ---- \imgN%d.jpg"%count,image)
        cv2.imwrite(r"C:\Users\suruj\PYTHON-IMAGE-PROCESSING\RNF (read n fetch) API\FRAME1\imgN%d.jpg"%count,image)
        vidcap.set(cv2.CAP_PROP_POS_MSEC,(count**100)) #SETTING THE SPEED
        ret,image = vidcap.read()
        cv2.imshow("frame",image)
        #print(count)
        count +=1
        if cv2.waitKey(1) == ord("q"):
            break
            cv2.destroyAllWindows()

vidcap.release()
cv2.destroyAllWindows()
```

Thank You