**Iris Recognition**

Code:

import cv2

import numpy as np

import os

import pandas as pd

import time

start\_time=time.time()

test\_original = cv2.imread("input\_image.bmp")

cv2.imshow("Original", cv2.resize(test\_original, None, fx=1, fy=1))

cv2.waitKey(0)

cv2.destroyAllWindows()

count=0

x=False

for file in [file for file in os.listdir("final")]:

iris\_database\_image = cv2.imread("./final/" + file)

sift = cv2.xfeatures2d.SIFT\_create()

keypoints\_1, descriptors\_1 = sift.detectAndCompute(test\_original, None)

keypoints\_2, descriptors\_2 = sift.detectAndCompute(iris\_database\_image, None)

matches = cv2.FlannBasedMatcher(dict(algorithm=1, trees=10),

dict()).knnMatch(descriptors\_1, descriptors\_2, k=2)

match\_points = []

start\_time=time.time()

for p, q in matches:

if p.distance < 0.1 \* q.distance:

match\_points.append(p)

keypoints = 0

if len(keypoints\_1) <= len(keypoints\_2):

keypoints = len(keypoints\_1)

else:

keypoints = len(keypoints\_2)

if (len(match\_points) / keypoints) > 0.95:

print("The input Iris image is matched!!")

print("percentage(%) of match: ", len(match\_points) / keypoints \* 100)

b=time.time()-start\_time

print("the total time taken is :{} Seconds".format(b))

result = cv2.drawMatches(i, keypoints\_1, iris\_database\_image,

keypoints\_2, match\_points, None)

result = cv2.resize(result, None, fx=1, fy=1)

cv2.imshow("result", result)

cv2.waitKey(10000)

cv2.destroyAllWindows()

x=True

count+=1

break

if x:

break

else:

print("The input Iris image is not matched!!")

Input Iris Image:



Processing the Input Image:



**Output:**

If Iris is matched:



If Iris is not matched:



Description:

* The approach of this project involves how the minutia points are extracted from the iris images and after that the pattern matching process goes on
* It uses K-Nearest Neighbour (KNN) algorithm to train the dataset.
* This project uses Scale – Invariant Feature Transform (SIFT) algorithm to recognize the Iris.
* This project takes two both left and right iris as input, processes the both iris images at a time gives the output as
* Input Iris is matched
* No Iris is matched