USING DETECTION METHOD

FUNCTIONS USED IN THE PROGRAM

① dev(x, y);
returns the deviation of x from y.
how it works

1. Assuming format (img) == format (DB. img ())
2. Assuming the image is noise free.

② avg(arr[]);
returns average of an array

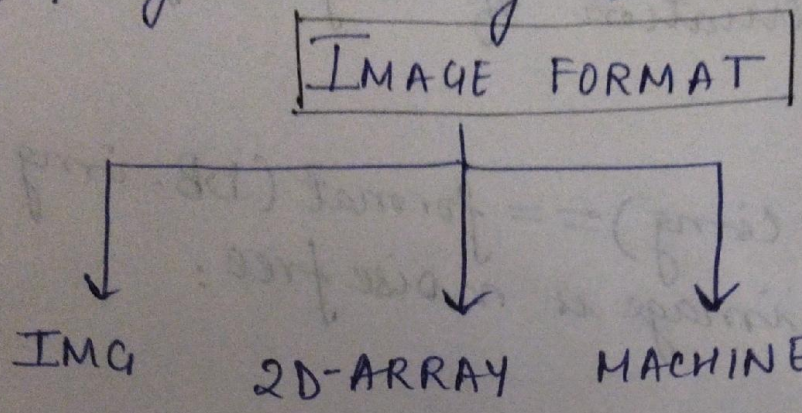
③ Format (x);
returns format of x;

④ Concat (x, y);
Adds x & y together

⑤ Convert (x, y);
returns converted form of 'x'. i.e. converts x into 'y'. file. where;
y ⇒ format
x ⇒ any format input

COMMONLY USED TERMS

Img ⇒ signature image extracted from cheque.



format(DB.img[ ] == Img)

NOTE: If the given condition is false, we would use (Img, format(DB.img[ ]))

DB.img[ ] ⇒ This would store 'n' number of signatures in the database, which acts as an input from the user.

ALGORITHM

```
{
dev_db; dev_img; avg_dev; x; //arbitrary variable
flag = 0;
functions()
for(i => 0 to m)
    deviation_ar[i] = dev(DB_img[i], DB_img[varii]);
//deviation of ith element from rest all
//element is stored in the ith position of deviation_ar[];
avg_dev = avg(deviation_ar[n]);
// we have average deviation of Database
//stored signatures
//assume from format is matched after conversion using convert();
for( i => 0 to n)
    if(dev(img, DB_img[i]) >
        (avg.dev + x))
            flag = 1;
            break;
// sign is forged now
}
```

A.I. GENERATION OF SIGNATURES

FUNCTIONS USED

- ①. A.generate(a, x);

generates 'a' no. of signatures deviating ((±x) ⇒ range)% from its parent 'A'.

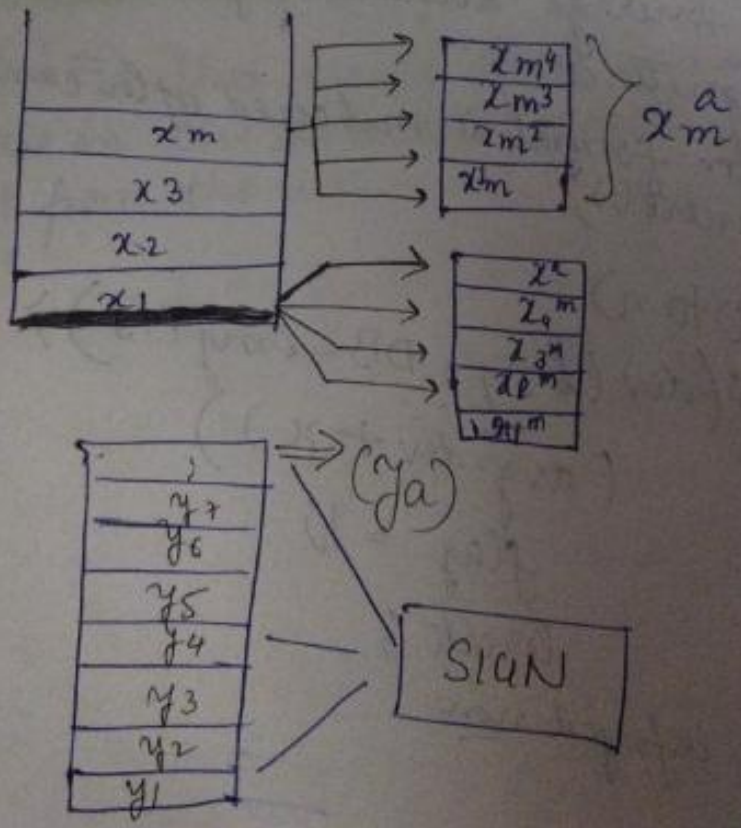
TERMS USED

① SIFT

Scale Invarient Feature Transform. Computer vision algorithm to detect, describe and match local features in image.

PSEUDOCODE ~~FLOW~~ DIAGRAM

AI(img, DB_img[])



NOTE: Here,

format(img) == format(DB_image)

METHOD

* 'a' number of AI generated signatures having very less deviation from sign X_i in database

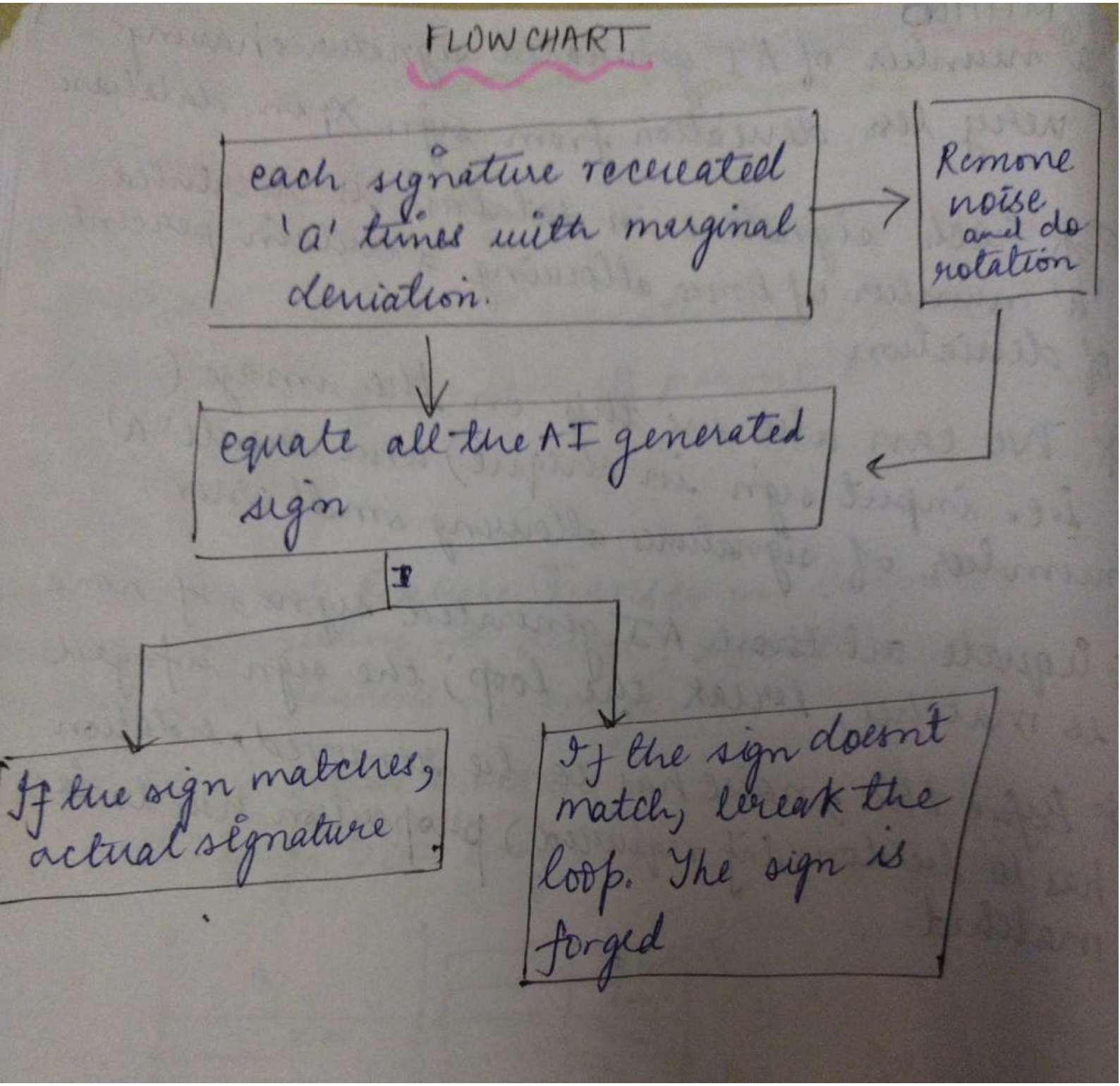
* Each signature in database is recreated 'a' number of times allowing a certain percent of deviation

* We can also use this on the image (i.e. input sign in cheque) and create 'a' number of signatures allowing small error.

* Equate all those AI generated signs, if none is matches, break the loop; the sign is forged

* Before this, noise has to be removed, rotation has to be done, (if required) proportion has to be matched.

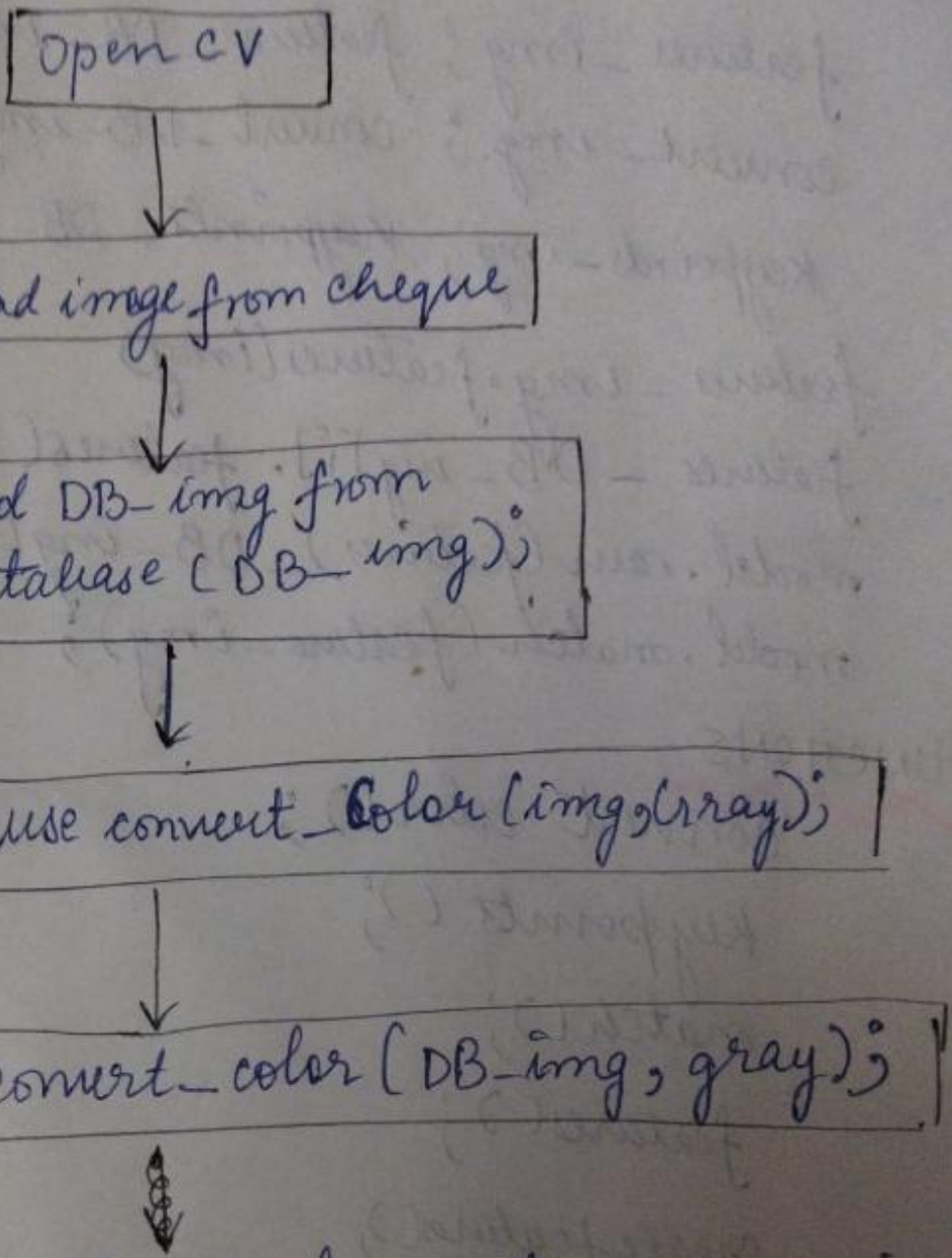
FLOW CHART


each signature recreated 'a' times with marginal deviation.
Remove noise and do rotation
equate all the AI generated sign
If
If the sign matches, actual signature
If the sign doesn't match, break the loop. The sign is forged

```
for (i=0 to n  and flag==0) {
    for (j=0 to a) { // check if signs matches or not
        if (match(AI_DB_ar[i][j],
                AI_unknown_ar[i]) {
            for  flag=1
            break ;  // sign is real
        }
    }
}
```

USING SIFT, SVM AND OPENCV

PSEUDOCODE



NOTE: ① Here have to store and convert every DB image and store into another array and store into another array.

② store key points using img, DB_img into x_1 & x_2

ALGORITHM

// img → cheque sign image
// DB_img[ ] → array of database images,
features_img ; features_DB_img[ ];
convert_img ; convert_DB_img[ ];
keypoints_img ; keypoints_DB_img[ ];
features_img.features(img)
features_DB_img[i].features(DB_img[i])
model.train(features)_DB_img[i]
model.match(feature_img);

FUNCTIONS

convert color( );
keypoints ( );
match( );
features( );
createfeatures( );
predict( );