**MINI PROJECT- II REPORT**

**On**

**SKETCH BASED SUSPECT RETRIEVAL**

**Submitted by**

AKASH AGRAWAL

171500025

DIVYA AGRAWAL

171500101

DIVYA SINGHAL

171500102

**Under the Guidance of**

Mr. Mandeep Singh

(Technical Trainer)

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**2020**

**Abstract**

The problem of matching a sketch to a gallery of suspect images is addressed in this project. A sketch drawn by a forensic artist is processed to define it as a concatenation of features; an image vector. Each image in the gallery is then iterated through and processed similarly to find the closest match.

Face sketches have the essential information abut the spatial topology and geometric details of faces while missing some important facial attributes such as ethnicity, hair, eye and skin colour.

The proposed architecture is able to make full use of the sketch and complementary facial attribute information to train a machine learning model compared to the conventional sketch-photo recognition methods.

**Acknowledgement**

It is my pleasure to acknowledge the assistance of a number of people without whose help this project would not have been possible.

This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

First and Foremost, I would like to express our gratitude to **Mr. Mandeep Singh (Technical Trainer)** my project guide, for providing invaluable encouragement, guidance and assistance. I would like to thank lab workers for the operation extended to us throughout the project. After doing this project I can confidently say that this experience has not only enriched me with technical knowledge but also has upraised the maturity of thoughts and vision. The attributes required being a successful professional.

We also want to have an opportunity to acknowledge the contribution of all the faculty members of the department for their guidance and cooperation throughout the project.

**Name of Candidate:** Akash Agrawal

**Roll. No.:** 171500025

**Name of Candidate:** Divya Agrawal

**Roll. No.:** 171500101

**Name of Candidate:** Divya Singhal

**­­Roll. No.:** 171500102



**Department of Computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**

**Declaration**

We hereby declare that the work which is being presented in the Mini Project “**SKETCH BASED SUSPECT RETRIEVAL**”**,** is partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering of GLA University, Mathura, is an authentic record of my own work carried under the supervision of **Mr. Mandeep Singh.**

**Name of Candidate:** Akash Agrawal

**Roll. No.:** 171500025

**Name of Candidate:** Divya Agrawal

**Roll. No.:** 171500101

**Name of Candidate:** Divya Singhal

**Roll. No.:** 171500102

Contents

[Project Overview 1](#_Toc38464258)

[**1.1.** **Introduction and Motivation** 1](#_Toc38464259)

[**1.2.** **Existing System** 2](#_Toc38464260)

[**1.3.** **Proposed System** 2](#_Toc38464261)

[**1.3.1.** **SURF** 3](#_Toc38464262)

[**1.3.2.** **SVM (Support Vector Machine)** 4](#_Toc38464263)

[Project Design 7](#_Toc38464264)

[**2.1** **EXPLANATION OF BLOCK DIAGRAM** 8](#_Toc38464265)

[**2.1.1. Input Image** 8](#_Toc38464266)

[**2.1.2. Input Sketch** 8](#_Toc38464267)

[**2.1.3 Pre Processing** 8](#_Toc38464268)

[References 9](#_Toc38464269)

**CHAPTER- 1**

# Project Overview

* 1. **Introduction and Motivation**

Humans have an inherent sense for justice, developed at infancy and nurtured though society from then on. So when one witnesses a crime happening, his/her instinct is to want to stop it. And after the crime, one wishes to see the criminal punished. Individuals who have been at the scene of the crime can report to appropriate law enforcement agencies about the criminal. This process is often complicated by the fact that witnesses to crimes cannot identify the individual; except through his face.

Sketch to image matching is a topic that has been in discussion for the past decade. In present scenario, we are dealing with manual searching from thousands of criminal records present in record room. Any automated system for the same purpose is not in existence yet. However, most algorithms proposed have been used earlier in face recognition systems, such as SIFT and SURF. The major challenge for this task is overcoming the abstractness or inaccuracy of a sketch. Like beauty being in the eye of the beholder, the description of and interpretation of a facial feature varies with the witness's memory and the artist's skill. Some sketches are purposely abstract so as to draw attention to its few concrete features.

As this research is purely technical, these limitations can be ignored, and an attempt is made to match a sketch across a gallery of images to that which best fits it.

* 1. **Existing System**

In present scenario, we are dealing with manual searching from thousands of records. Any automated system for the same purpose is not in existence yet.

* 1. **Proposed System**

In order to reduce the manual work of searching the suspect from thousands of records present in record room as it takes a lot of time of around ten to fifteen days, this system tends to search the suspect with the help of sketch with just a single click.

The process of the proposed system is described below.

* Images are retrieved from the gallery one by one and matched with a given sketch
* The most likely suspect images are output to the user.
* The eye-witness is asked to identify the alleged criminal. This can result in two outcomes:
* The criminal is identified. Here the sketch matching process ends.
* The criminal is not among the resultant images. Here, the result images are still qualified as suspects, and optionally, the sketch can be modified to repeat the matching process.

Brief descriptions of the two implemented algorithms are provided in the section below:

* + 1. **SURF**

SURF stands for Speeded Up Robust Features. It is an approach which is generally used to construct a robust image feature detector and descriptor. It can be used in computer vision tasks like object recognition and 3D reconstruction. The main interest of the SURF approach lies in its fast computation of operators using box filters, thus enabling real-time applications such as tracking and object recognition.

SURF is composed of two steps

--> Feature Extraction

--> Feature Description

**Detection**

SURF uses square-shaped filters as an approximation of Gaussian smoothing. Filtering the image with a square is much faster if the integral image is used:

**S(x, y) =**

In order to detect the interest points, SURF uses a blob detector based on the Hessian matrix. Blob like structures are detected in the image, where the local determinant of the Hessian matrix is maximum.

Given a point p=(x, y) in an image I, the Hessian matrix H(p, σ) at point p and scale σ, is:

**H(p, σ) = Lxx (p, σ) Lxy (p, σ)**

**Lyz (p, σ) Lyy (p, σ)**

where Lxx(p, σ) etc. is the convolution of the second-order derivative of Gaussian with the image I(x, y) at the point p.

**Descriptors**

The goal of a descriptor is to provide a unique and robust description of an image feature, e.g., by describing the intensity distribution of the pixels within the neighbourhood of the point of interest. Most descriptors are thus computed in a local manner, hence a description is obtained for every point of interest identified previously.

To extract the descriptors, "sum of Haar wavelet responses" is used. We first construct a square region centred at the interest point and oriented along the orientation decided by a special selection method. We then use these descriptors to describe any given image and compare these descriptors to give us the similarity between sketch and image.

The dimensionality of the descriptor has direct impact on both its computational complexity and point-matching robustness/accuracy. In addition, only 64 dimensions are usually used in SURF to reduce the time cost for both feature computation and matching. Because each of SURF feature has only 64 dimensions in general and an indexing scheme is built by using the sign of the Laplacian, SURF is much faster.

* + 1. **SVM (Support Vector Machine)**

A support vector machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression problems. SVMs are most commonly used in classification.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

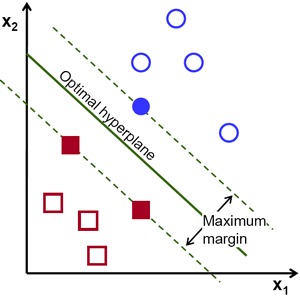


Fig: 1.1 Optimal Hyperplane

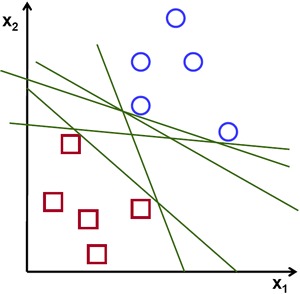


Fig: 1.2 Possible Hyperplane

* **Hyperplanes** are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. Therefore, dimension of hyperplane is equals to number of input features minus one It becomes difficult to imagine when the number of features exceeds 3.
* **Support vectors** are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. These are the points that help us build our SVM.
* It can be extended to patterns that are not linearly separable by transformations of original data to map into new space using the kernel functions
* SVMs are fast because it uses kernel trick in dual problems
* Kernel search an optimal separating hyperplane in a higher dimensional space without increasing the computational complexity much. It can be applied if the algorithm takes the features in terms of inner product.
* According to duality principle, optimization problems may be viewed as 'primal' or 'dual'. For a convex optimization problem, the primal and dual have same optimum solution.

**Advantage:-**

* SVMs are very good when we have no idea on the data.
* Works well with even unstructured and semi structured data like text, images and trees.
* The kernel trick is real strength of SVM. With an appropriate kernel function, we can solve any complex problem
* SVM is not solved for local Optima
* It scales relatively well to high dimensional data
* SVM model have generalization in practice. The risk of over-fitting is less in SVM.

**CHAPTER- 2**

# Project Design

This chapter presents the plan of the undertaking. An info sketch is taken care of into the framework where it experiences certain measure of pre-preparing, for example, transformation to highly contrasting. The sketch in this way got, experiences a calculation (SIFT or SURF) where it experiences include extraction as descriptors. These descriptors are then spared. For each test picture/photograph in the exhibition, the equivalent previously mentioned process is rehashed with each and their descriptors are spared. Separation between each pair of sketch-photograph descriptors is processed according to closest neighbor and dependent on a likeness measure, for example, number of matches for SIFT or highlight separation if there should be an occurrence of SURF, photographs are retrived in the order for the best to worst match.

Pre processing

Array of Feature Values

Feature Extraction

Input Image

Matched Sketch-Image Pair

Matching Algorithm

Distance Calculation

Array of Feature Values

Feature Extraction

Pre processing

Input Sketch

Fig 2.1 Block Diagram of Sketch To Image Matching

* 1. **EXPLANATION OF BLOCK DIAGRAM**

### **2.1.1. Input Image**

The Image gallery is iterated through to get each picture.

### **2.1.2. Input Sketch**

The input 2D, Black and white sketch is given as input to the system.

### **2.1.3 Pre Processing**

Pre-processing procedure consists of rescaling the image to the sketch size and converting the resultant to black and white image.

**2.1.4 Feature Extraction**

Extraction of features from the sketch is achieved by using any one of the sketch matching algorithms (i.c. SIFT or SURF). In this step, the features are identified, defined and then concatenated to give an image vector which can be used for understanding and comparing the image. Feature extraction is carried on the sketch to find feature values. Similarly, the test image is pre-processed before its features being extracted and feature values being calculated.

**2.1.5 Array of Feature Values**

The resultant values are stored of sketch and input images are obtained and stored in their respective arrays.

**2.1.6 Distance Calculation**

The values obtained from the input sketch and images are compared with each other and the best fit image is returned.

**CHAPTER- 3**

# Implementation of Code

# 

import glob

import os

import cv2

import matplotlib.pyplot as plt

import numpy as np

photo\_dir="dataset/training/photo"

photo\_data\_path=os.path.join(photo\_dir, '\*g')

photo\_files=glob.glob(photo\_data\_path)

photos=[]

for f1 in photo\_files:

img=cv2.imread(f1)

photos.append(img)

sketch\_dir="dataset/training/sketch"

sketch\_data\_path=os.path.join(sketch\_dir, '\*g')

sketch\_files=glob.glob(sketch\_data\_path)

sketches=[]

for f1 in sketch\_files:

img=cv2.imread(f1)

sketches.append(img)

photo\_keypoints=[]

photo\_descriptor=[]

sketch\_keypoints=[]

sketch\_descriptor=[]

for i,j in zip(photos,sketches):

training\_image = cv2.cvtColor(i, cv2.COLOR\_BGR2RGB)

training\_gray = cv2.cvtColor(training\_image, cv2.COLOR\_RGB2GRAY)

test\_image = cv2.cvtColor(j, cv2.COLOR\_BGR2RGB)

test\_gray = cv2.cvtColor(test\_image, cv2.COLOR\_RGB2GRAY)

## Detect keypoints and Create Descriptor

surf = cv2.xfeatures2d.SURF\_create(800)

train\_keypoints, train\_descriptor = surf.detectAndCompute(training\_gray, None)

photo\_keypoints.append(train\_keypoints)

photo\_descriptor.append(train\_descriptor)

test\_keypoints, test\_descriptor = surf.detectAndCompute(test\_gray, None)

sketch\_keypoints.append(test\_keypoints)

sketch\_descriptor.append(test\_descriptor)

#print(photo\_keypoints)

# References

[1] <https://journals.sagepub.com/doi/10.1177/146135570200400102>

[2] <https://ieeeexplore.ieee.org/document/8698564>

[3] [https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=deep%20sk etch-%20photo%20face%20recognitionz](https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=deep%20sk%20%20%20%20%20etch-%20photo%20face%20recognitionz)

[4] <http://mmlab.ie.cuhk.edu.hk/archive/facesketch.html>