A Project report on

MISSING CHILD IDENTIFICATION

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

SNIGDHA BANDRI	(19H51A0558)
ADITHYA KUMACHALAM	(19H51A05K9)
TANVI KALVA	(19H51A05M0)

Under the esteemed guidance of

Mr. B.K Chinna Maddileti
(Asst. Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC & JNTUH, Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase-1 report entitled **"Missing child identification"** being submitted by Snigdha bandari (19H51A0558), Adithya kurmachalam (19H51A05K9), Tanvi kalva (19H51A05M0) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. B.K Chinna Maddileti
Asst. Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mr. B.K Chinna Maddileti,** Asst. Professor, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

Snigdha bandari 19H51A0558
Adithya kurmachalam 19H51A05K9
Tanvi kalva 19H51A05M0

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

In India a countless number of children are reported missing every year. Among the missing child cases a large percentage of children remain untraced. A novel use of deep learning methodology for identifying the reported missing child from the photos of multitude of children available, with the help of face recognition. The public can upload photographs of suspicious child into a common portal with landmarks and remarks. The photo will be automatically compared with the registered photos of the missing child from the repository. Classification of the input child image is performed and photo with best match will be selected from the database of missing children. For this, a deep learning model is trained to correctly identify the missing child from the missing child image database provided, using the facial image uploaded by the public. The Convolutional Neural Network (CNN), a highly effective deep learning technique for image-based applications is adopted here for face recognition. Face descriptors are extracted from the images using a pre-trained CNN model VGG-Face deep architecture. Compared with normal deep learning applications, our algorithm uses convolution network only as a high-level feature extractor and the child recognition is done by the trained SVM classifier. Choosing the best performing CNN model for face recognition, VGG-Face and proper training of it results in a deep learning model invariant to noise, illumination, contrast, occlusion, image pose and age of the child and it outperforms earlier methods in face recognition based missing child identification.

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

India is the second populous country in the world and children represent a significant percentage of total population. But unfortunately, a large number of children go missing every year in India due to various reasons including abduction or kidnapping, run-away children, trafficked children and lost children. A deeply disturbing fact about India's missing children is that while on an average 174 children go missing every day, half of them remain untraced. Children who go missing may be exploited and abused for various purposes. It is difficult to find the missing child and it take much time to find the missing child

1.1 PROBLEM STATEMENT

This project is to describe the concept of identifying missing children by using Deep Learning and Multiclass SVM classifier and to implement this project author has used below modules
Using public dataset of missing children called FGNET is used to train deep learning CNN prediction model. After training model whenever, public uploads any suspected child image then this model will check in trained model to detect whether this child is in missing database or not. This detected result will store in database and whenever want official persons will login and see that detection result.
SVM Multiclass classifier use to extract face features from images based on age and other facial features and then this detected face will input to CNN model to predict whether this face child exists in image database or not.

1.2 RESEARCH OBJECTIVE

This project is to describe the concept of identifying missing children by using Deep Learning and Multiclass SVM classifier and to implement this project author has used below modules Using public dataset of missing children's called FGNET is used to train deep learning CNN prediction model. After training model whenever, public uploads any suspected child image then this model will check in trained model to detect whether this child is in missing database or not. This detected result will store in database and whenever want official persons will login and see that detection result.

SVM Multiclass classifier use to extract face features from images based on age and other facial features and then this detected face will input to CNN model to predict whether this face child exists in image database or not.

1.3 PROJECT SCOPE AND LIMITATIONS

In India a countless number of children are reported missing every year. Among the missing child cases a large percentage of children remain untraced. The public can upload photographs of suspicious child into a common portal with landmarks and remarks. The photo will be automatically compared with the registered photos of the missing child from the repository. The Convolutional Neural Network (CNN), a highly effective deep learning technique for image based applications is adopted here for face recognition. The classification performance achieved for child identification system is 99.41%. It was evaluated on 43 Child cases.

Limitations:

To assess the flexibility of face recognition deep architecture against variations in image quality, artificially degraded images are created. Images obtained by changing noise level, brightness, contrast, lighting conditions, obstructions, blur, aspect ratio and face positions are used for testing the child identification system. Face identification accuracy is computed as the ratio of correctly identified face images to the total number of child face images in the test set.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1. HOG : HISTOGRAM OF ORIENTED GRADIENTS

2.1.1. INTRODUCTION

Histogram of Oriented Gradients, also known as HOG, is a feature descriptor. It is used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in the localized portion of an image. This method is quite similar to Edge Orientation Histograms and Scale Invariant Feature Transformation (SIFT). The HOG descriptor focuses on the structure or the shape of an object. It is better than any edge descriptor as it uses magnitude as well as angle of the gradient to compute the features. For the regions of the image, it generates histograms using the magnitude and orientations of the gradient.

2.1.2. MERITS, DEMERITS AND CHALLENGES

MERITS:

HOG is still advantageous for objection detection in the computer vision industry because of its computation speed and accuracy. HOG is widely used for the detection of pedestrians and medical image analysis. Therefore, HOG can be used to detect small-scaled images with less computational power, which means you can run HOG without having a powerful GPU.

DEMERITS:

One of the drawbacks of HOG is that its computation speed is tardy while detecting an object for large-scaled images as it uses a sliding window technique to extract features from every pixel of an image. Hence, the accuracy is not highly reliable compared to the current convolutional neural networks.

The disadvantage is that the final descriptor vector grows larger, thus taking more time to extract and to train using a given classifier.

Overall, there are several notable findings in this work. The fact that HOG greatly out-performs wavelets and that any significant degree of smoothing before calculating gradients damages the HOG results emphasizes that much of the available image information is from abrupt edges at fine scales, and that blurring this in the hope of reducing the sensitivity to spatial position is a mistake. Instead, gradients should be calculated at the finest available scale in the current pyramid layer, rectified or used for orientation voting, and only then blurred spatially. Given this, relatively coarse spatial quantization suffices (8×8 pixel cells / one limb width). On the other hand, at least for human detection, it pays to sample orientation rather finely: both wavelets and shape contexts lose out significantly here. Secondly, strong local contrast normalization is essential for good results. Better results can be achieved by normalizing each element (edge, cell) several times with respect to different local supports, and treating the results as independent signals. In our standard detector, each HOG cell appears four times with different normalizations.

2.1.3. IMPLEMENTATION OF HISTOGRAM OF ORIENTED GRADIENTS

In the case of HOG feature descriptors, we also convert the image (width x height x channels) into a feature vector of length *n* chosen by the user. Although it may be hard to view these images, these images will be perfect for image classification algorithms like SVMs in order to produce good results.

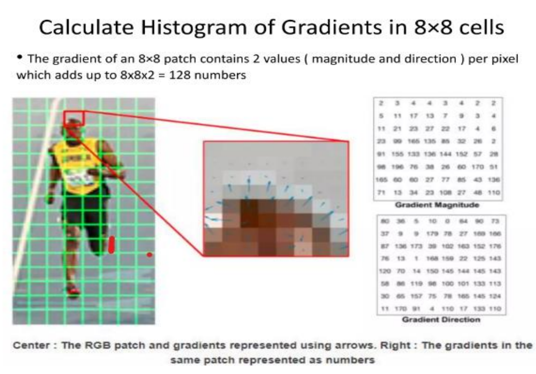


Fig. 2.1.3

Importing Libraries

```
from skimage.io import imread
from skimage.transform import resize
from skimage.feature import hog
from skimage import exposure
import matplotlib.pyplot as plt
```

Reading the image

```
img = imread('B.jpg')
plt.axis("off")
plt.imshow(img)
print(img.shape)
```

```
(781, 794, 3)
```

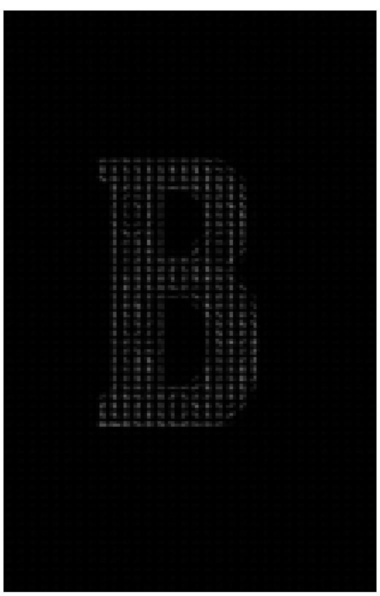
Resizing image

```
resized_img = resize(img, (128*4, 64*4))
plt.axis("off")
plt.imshow(resized_img)
print(resized_img.shape)
```

```
(512, 256, 3)
```

Creating and visualizing HOG Features

```
fd, hog_image = hog(resized_img, orientations=9, pixels_per_cell=(8, 8),
                        cells_per_block=(2, 2), visualize=True, multichann
plt.axis("off")
plt.imshow(hog_image, cmap="gray")
plt.show()
```



2.2. SURF: SPEED-UP ROBUST FEATURES

2.2.1. INRODUCTION

The task of finding point correspondences between two images of the same scene or object is part of many computer vision applications. Image registration, camera calibration, object recognition, and image retrieval are just a few. It utilizes the integral image, hessian matrix and haar wavelet responses to improve the performance in robust way.

To detect interest points, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed with 3 integer operations using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest.

2.2.2. MERITS, DEMERITS

MERITS:

- One big advantage of this approximation is that, convolution with box filter can be easily calculated with the help of integral images. And it can be done in parallel for different scales. Also the SURF rely on determinant of Hessian matrix for both scale and location.

- SURF adds a lot of features to improve the speed in every step. Analysis shows it is 3 times faster than SIFT while performance is comparable to SIFT. SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change.

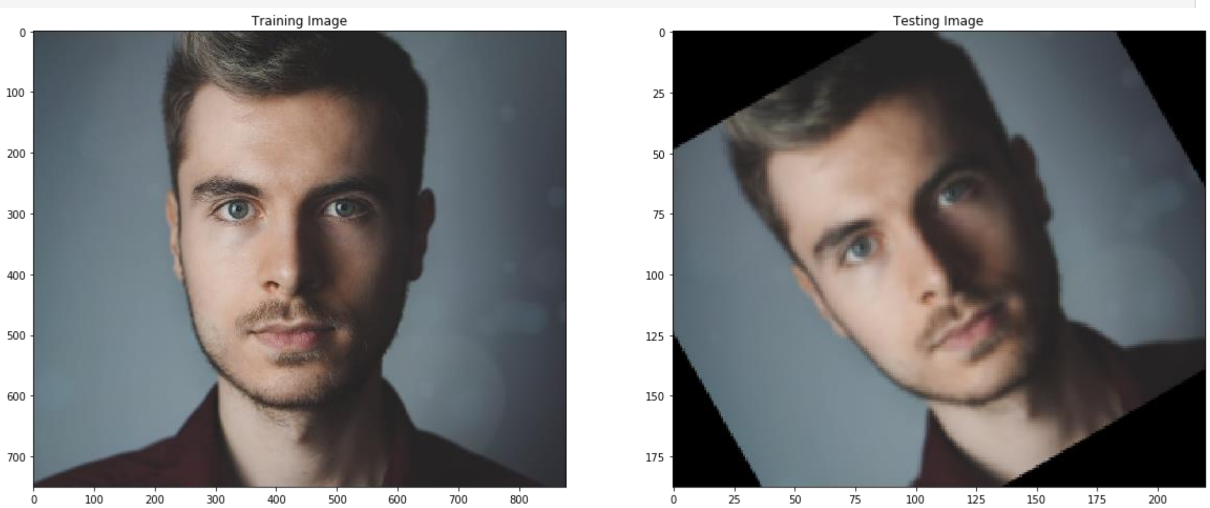
- Another major advantage of SURF is its low computation time: detection and description of 1529 interest points takes about 610 ms, the upright version U-SURF uses a mere 400 ms.

DEMERITS:

- Poor image quality limits facial recognition’s effectiveness
- Small images sizes make facial recognition more difficult
- Different face angles can throw off facial recognition’s reliability
- Data Processing and storage can limit facial recognition technology

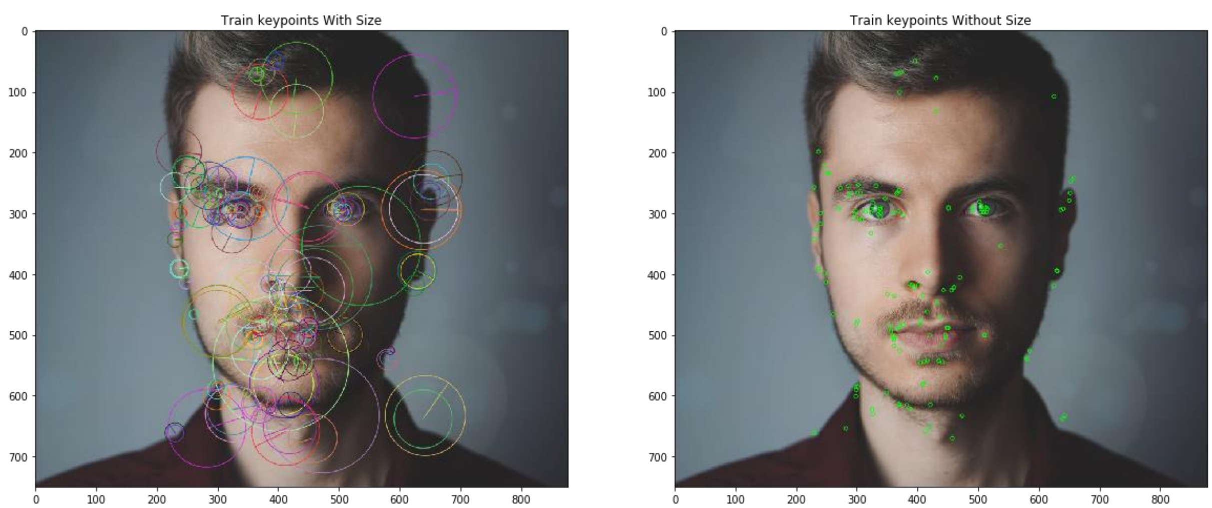
2.2.3. IMPLIMENTATION

```
import cv2
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline

# Load the image
image1 = cv2.imread('./images/face1.jpeg')

# Convert the training image to RGB
training_image = cv2.cvtColor(image1, cv2.COLOR_BGR2RGB)

# Convert the training image to gray scale
training_gray = cv2.cvtColor(training_image, cv2.COLOR_RGB2GRAY)

# Create test image by adding Scale Invariance and Rotational Invariance
test_image = cv2.pyrDown(training_image)
test_image = cv2.pyrDown(test_image)
num_rows, num_cols = test_image.shape[:2]

rotation_matrix = cv2.getRotationMatrix2D((num_cols/2, num_rows/2), 30, 1)
test_image = cv2.warpAffine(test_image, rotation_matrix, (num_cols, num_ro

test_gray = cv2.cvtColor(test_image, cv2.COLOR_RGB2GRAY)

# Display traning image and testing image
fx, plots = plt.subplots(1, 2, figsize=(20,10))

plots[0].set_title("Training Image")
plots[0].imshow(training_image)
```



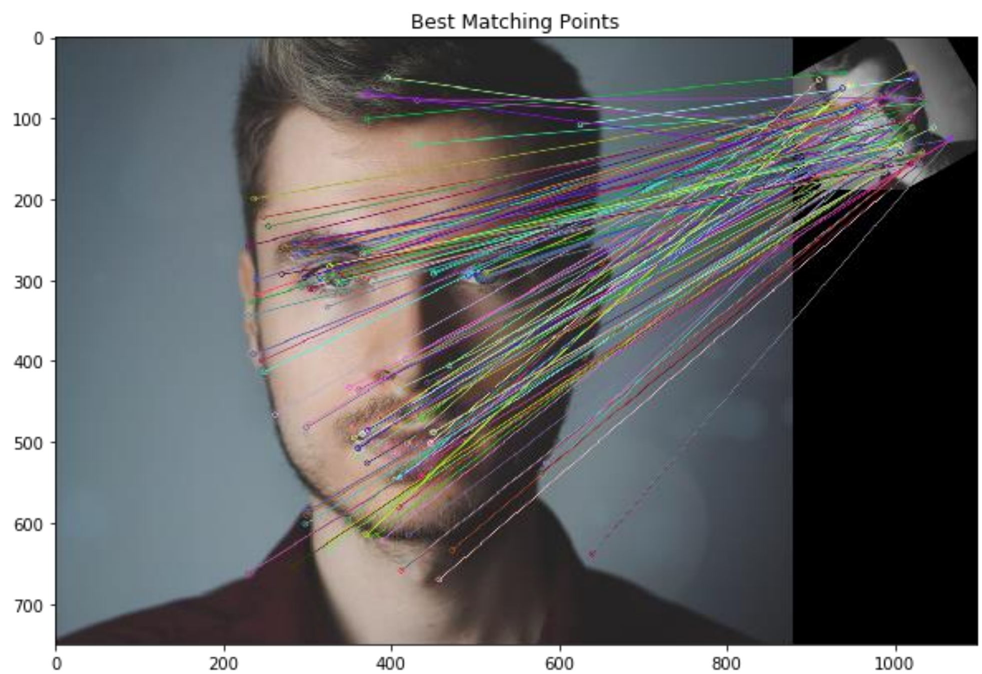
Detect keypoints and Create Descriptor

```
surf = cv2.xfeatures2d.SURF_create(800)

train_keypoints, train_descriptor = surf.detectAndCompute(training_gray, N
test_keypoints, test_descriptor = surf.detectAndCompute(test_gray, None)

keypoints_without_size = np.copy(training_image)
keypoints_with_size = np.copy(training_image)

cv2.drawKeypoints(training_image, train_keypoints, keypoints_without_size,

cv2.drawKeypoints(training_image, train_keypoints, keypoints_with_size, fl

# Display image with and without keypoints size
fx, plots = plt.subplots(1, 2, figsize=(20,10))

plots[0].set_title("Train keypoints With Size")
plots[0].imshow(keypoints_with_size, cmap='gray')

plots[1].set_title("Train keypoints Without Size")
plots[1].imshow(keypoints_without_size, cmap='gray')

# Print the number of keypoints detected in the training image
print("Number of Keypoints Detected In The Training Image: ", len(train_ke

# Print the number of keypoints detected in the query image
print("Number of Keypoints Detected In The Query Image: ", len(test_keypoi
```



Matching Keypoints

```
# Create a Brute Force Matcher object.
bf = cv2.BFMatcher(cv2.NORM_L1, crossCheck = False)

# Perform the matching between the SURF descriptors of the training image
matches = bf.match(train_descriptor, test_descriptor)

# The matches with shorter distance are the ones we want.
matches = sorted(matches, key = lambda x : x.distance)

result = cv2.drawMatches(training_image, train_keypoints, test_gray, test_

# Display the best matching points
plt.rcParams['figure.figsize'] = [14.0, 7.0]
plt.title('Best Matching Points')
plt.imshow(result)
plt.show()

# Print total number of matching points between the training and query ima
print("\nNumber of Matching Keypoints Between The Training and Query Image
```



2.3. SIFT: SCALE INVARIANT FEATURE TRANSFORM

2.3.1. INTRODUCTION

SIFT stands for Scale-Invariant Feature Transform and was first presented in 2004, by **D.Lowe**, University of British Columbia. SIFT is invariance to image scale and rotation. This algorithm is patented, so this algorithm is included in the Non-free module in OpenCV.

SIFT can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling, orientation, illumination changes, and partially invariant affine distortion. This section summarizes the original SIFT algorithm and mentions a few competing techniques available for object recognition under clutter and partial occlusion.

2.3.2. MERITS AND DEMERITS

MERITS:

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
- **Distinctiveness:** individual features can be matched to a large database of objects
- **Quantity:** many features can be generated for even small objects
- **Efficiency:** close to real-time performance
- **Extensibility:** can easily be extended to a wide range of different feature types, with each adding robustness

DEMERITS:

- Generally, the high dimensionality of the descriptor is a drawback of SIFT at the matching step. For on-line applications relying only on a regular PC, each one of the three steps (detection, description, matching) has to be fast.
- SIFT has been reported to be 3 times slower than SURF

2.3.3. IMPLIMENTATOIN

```
import cv2

# read the images
img1 = cv2.imread('book.jpg')
img2 = cv2.imread('table.jpg')
# convert images to grayscale
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2GRAY)
# create SIFT object
sift = cv2.xfeatures2d.SIFT_create()
# detect SIFT features in both images
keypoints_1, descriptors_1 = sift.detectAndCompute(img1,None)
keypoints_2, descriptors_2 = sift.detectAndCompute(img2,None)
```

The above code loads the image and convert it to grayscale, let's create SIFT feature extractor object:

```
# create SIFT feature extractor
sift = cv2.xfeatures2d.SIFT_create()
```

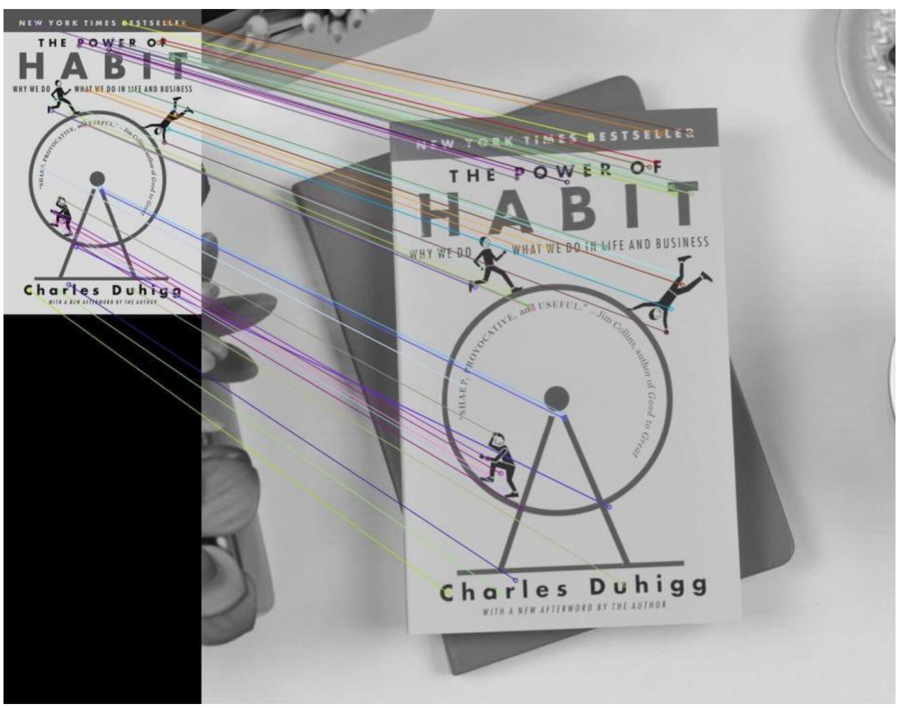
To detect the keypoints and descriptors, we simply pass the image to `detectAndCompute()` method:

```
# detect features from the image
keypoints, descriptors = sift.detectAndCompute(img, None)
```

Finally, let's draw the keypoints, show and save the image:

```
# draw the detected key points
sift_image = cv2.drawKeypoints(gray, keypoints, img)
# show the image
cv2.imshow('image', sift_image)
# save the image
cv2.imwrite("table-sift.jpg", sift_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

OUTPUT:



CHAPTER 3
RESULTS AND DISCUSSION

CHAPTER 3

RESULTS AND DISCUSSION

3.1. COMPARISION OF EXISTING SOLUTIONS

- HOG AND SURF:

TABLE III. 5-Scale Testing with SURF Extraction and HOG Extraction

SURF		
Known	***Parasitized***	***Uninfected***
Parasitized	81%	19%
Uninfected	4%	96%
Average Accuracy	88.50%	
HOG		
Known	***Parasitized***	***Uninfected***
Parasitized	79%	21%
Uninfected	6%	94%
Average Accuracy	86.50%	

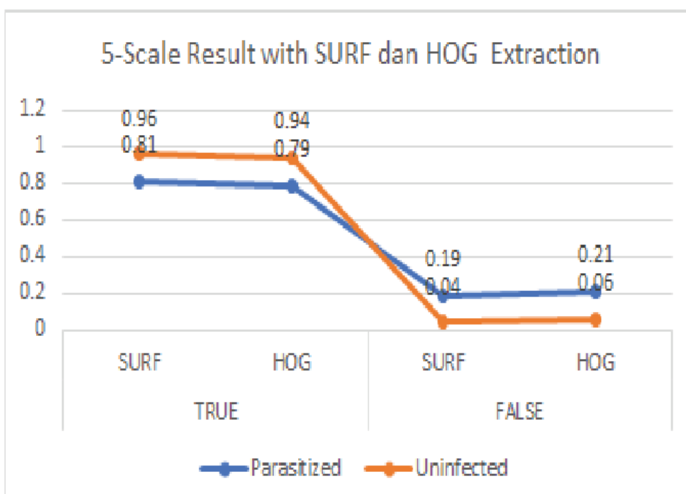


Fig. 3.1

- SIFT AND SURF:

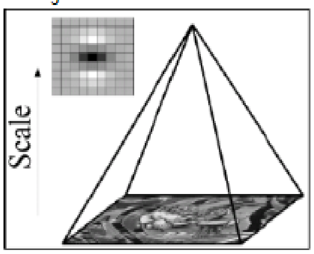
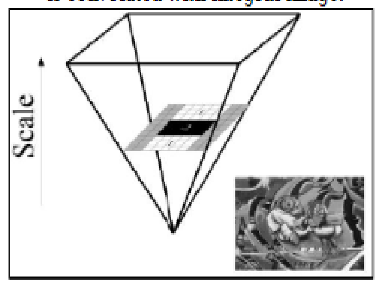
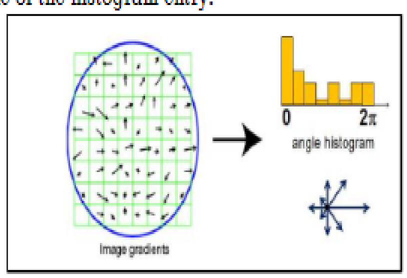
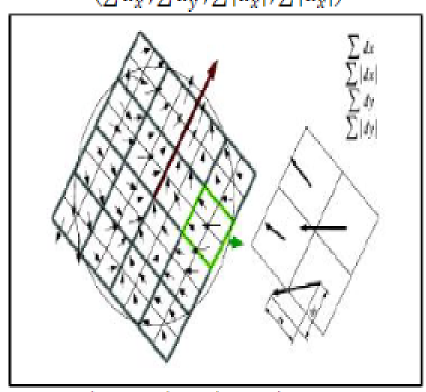
	SIFT	SURF
Scale Space	Difference of Gaussian (DoG) is convolved with different size of images with same size of filter.  Fig. 6. Fix filter is convoluted with down sampling images	Different size of box filter(Laplacian of Gausian (LoG)) is convoluted with integral image.  Fig. 7. Fix image is convoluted with up sampling filters.
Key point detection	Using of local extrema detection, apply Non maxima suppression and eliminate edge response with Hessian matrix	Determine the key points with Hessian matrix and Non Maxima suppression.
Orientation	Image gradiant magnitude and orientations are sampled around the key point location, using the scale of the key point to select the level of Gaussian blur for the image. Orientation of histogram is used for same.	A sliding Orientation window of size $\pi/3$ detects the dominant orintation of the Gaussian weighted Haar Wavelet responses at every sample point with in a circular neighbourhood around the interest points.
Descriptor	The key point descriptor allows for significant shift in gradiant positions by creating orientation histograms over 4 x 4 sample regions. The figure shows 8 directions for each orientation histogram, with the length of each arrow corresponding to the magnitude of the histogram entry.  Fig. 8. Orientation assignment	An orientation quadratic grid with 4x4 square sub regions is laid over the interest point. For each square, the wavelet responses are computed from 5x5 samples. Descriptor of SURF is $V = (\sum d_x, \sum d_y, \sum \lvert d_x \rvert, \sum \lvert d_x \rvert)$  Fig. 9: Orientation assignments
Size of descriptor	128 bits	64 bits

Fig. 3.1.1

3.2 . PERFORMANCE METRICS

To assess the flexibility of face recognition deep architecture against variations in image quality, artificially degraded images are created. Images obtained by changing noise level, brightness, contrast, lighting conditions, obstructions, blur, aspect ratio and face positions are used for testing the child identification system. Face identification accuracy is computed as the ratio of correctly identified face images to the total number of child face images in the test set.

$$Accuracy = \frac{Correctly\ recognized\ face\ images}{Total\ number\ of\ child\ face\ images}$$

The computed recognition accuracy of the multi class SVM using learned features from CNN is 99.41%.

Fig. 3.2

CHAPTER 4

CONCLUSION

A missing child identification system is proposed, which combines the powerful CNN based deep learning approach for feature extraction and support vector machine classifier for classification of different child categories. This system is evaluated with the deep learning model which is trained with feature representations of children faces. By discarding thesoftmax of the VGG-Face model and extracting CNN image features to train a multi class SVM, it was possible to achieve superior performance. Performance of the proposed system is tested using the photographs of children with different lighting conditions, noises and also images at different ages of children. The classification achieved a higher accuracy of 99.41% which shows that the proposed methodology of face recognition could be used for reliable missing children identification.

CHAPTER 5
REFERENCES

1.Rohit Satle, Vishnuprasad Poojary, John Abraham, Shilpa Wakode, "Missing child 2.identification using face recognition system", International Journal of Advanced Engineering and Innovative Technology (IJAEIT), Volume 3 Issue 1 July - August 2016.

3.https://en.wikipedia.org/wiki/FindFace

4. https://www.reuters.com/article/us-china-trafficking-apps/mobileapp-helps-china-recover-hundreds-of-missing-childrenidUSKBN15J0GU