Automated Fraud Detection Framework in Examination Halls

Anandhavalli D[1], Devi Natchiyar N[2], Deepshika H[3], Priyadharshini M[4]

[1]*Assistant Professor, Department of IT*
[2,3,4]*UG Students, Department of IT, Velammal College of Engineering and Technology, Madurai*

Abstract - *"Marks in exams doesn't always equal intelligence or brilliance". But, in today's world, there is a widespread belief that the marks which people score in their examinations determine their future. This assumption has led to a huge number of problems in the society such as traumatized youth and in worst cases has even led to suicide attempts. People have started resorting to unfair practices in order to clear an examination. These practices include impersonation of the candidate even by his/her own tutors at times and the usage of electronic gadgets to help them answer questions. The objective of this study was to develop an automated fraud detection framework to detect the impersonation of candidates and possession of electronic gadgets by the candidates in an examination hall. The impersonation of the candidates and the presence of electronic gadgets have been detected by image processing techniques for detection and recognition and machine learning algorithms for the classification. The machine learning algorithms specifically used in this study are the Random Forest algorithm and Histogram of Oriented Gradients (HoG) algorithm. These algorithms have been selected for the high accuracy that they provide in detecting and recognizing datasets. They require a smaller number of training datasets when compared to other models to perform efficiently.*

Key words: **Impersonation, Electronic Gadgets, Random Forest, Histogram of Oriented Gradients (HoG), Datasets, Face Detection, Face Recognition, Training.**

1. INTRODUCTION

Today, Examinations around the world are prevalently conducted in two modes – online and offline. The offline examinations make use of pen and paper while the online examinations are conducted with the help of computers and most often dedicated software testing systems. Even though every aspect of life today has started becoming computerized and digital, the examination committees still rely on human invigilators to ensure the proper conduct of candidates in an examination hall be it an online or offline examination.

For instance, the Intermediate Public Examinations conducted in Hyderabad in 2019 had nearly 37 cases of severe malpractices as a result of which those candidates were debarred from appearing for the remaining tests.

With cases like these, we require a dedicated and specialized system for monitoring the conduct of students in examination halls.

To reduce the human effort in invigilation of an examination hall and provide a highly efficient candidate monitoring system, we have proposed an automated fraud detection framework based on machine learning and image processing.

The Framework makes use of the machine learning algorithms called Random Forest algorithm and Histogram of Oriented Gradients (HoG) algorithm for detection and classification of datasets. These algorithms have been proved to provide highly accurate results when compared to other classification algorithm.

2. METHOD

2.1. Block Diagram

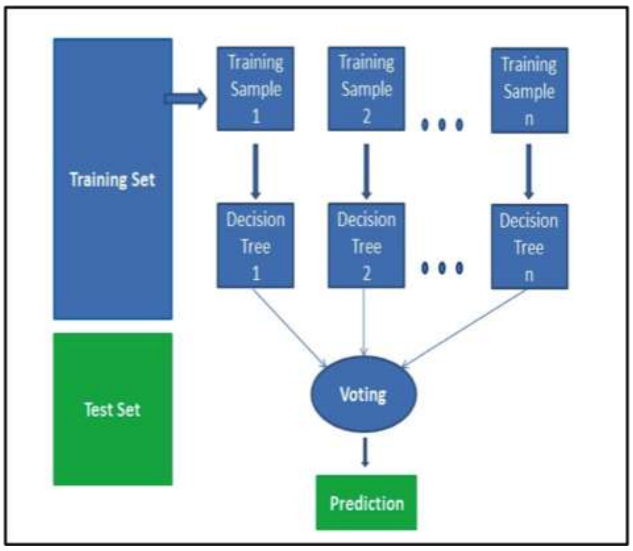


Fig -1: Block Diagram of Proposed System

The machine has been trained using a number of datasets provided as driving datasets. The process of training the machine and making it performance ready has been done in three steps.

- Initial Setup
- Dataset Collection
- Dataset Recognition

2.2. Initial Setup

- **Step 1:** Installation Of Ubuntu Operating System
- **Step 2:** Installation Of OpenCV

- **Step 3:** Installation Of the essential Python Packages.

2.2.1. Ubuntu in Machine Learning

Linux machines are better for machine learning. It is better for software development in general and you can find many flame wars on this. Ubuntu comes with better package management so it easier to install the common stuff. For the uncommon stuff it is far more likely to install or compile if needed on Linux with ease. Linux is more similar to your production machines. Also Ubuntu has a modern bash and modern GNU Sort.

2.2.2. OPENCV Vs MATLAB

Matlab is great for experimentation, but in OpenCV, implementation speed has been addressed very well, and on top of that you have the optimized compilation. For any production purposes, Matlab is not likely to be able to process data fast enough and cheaply enough.

2.2.3. Why Python?

- It is simple and consistent
- It has an extensive library and framework
- It is platform independent
- It is open source and has a great developers community and popularity.

2.3. Dataset Collection

- **Step 1:** Imported the numpy python dependency package.
- **Step 2:** Collected a few sets of driving images
- **Step 3:** Marked the anomalies in the driving images using numpy and saved them as the training data sets.

2.3.1. Numerical Python

NumPy (Numerical Python) is a linear algebra library in Python. It is a very important library on which almost every data science or machine learning Python packages such as SciPy (Scientific Python), Mat–plotlib (plotting library), Scikit-learn, etc depends on to a reasonable extent.

NumPy is very useful for performing mathematical and logical operations on Arrays. It provides an abundance of useful features for operations on n-arrays and matrices in Python.

2.4. Dataset Recognition

- **Step 1:** Train the system using the driving images obtained as the training data sets.
- **Step 2:** Acquire snapshots of the test scenario.
- **Step 3:** Use Histogram of Oriented Gradients (HOG) to process the input images and produce desired results with high accuracy.

3. ALGORITHMS USED

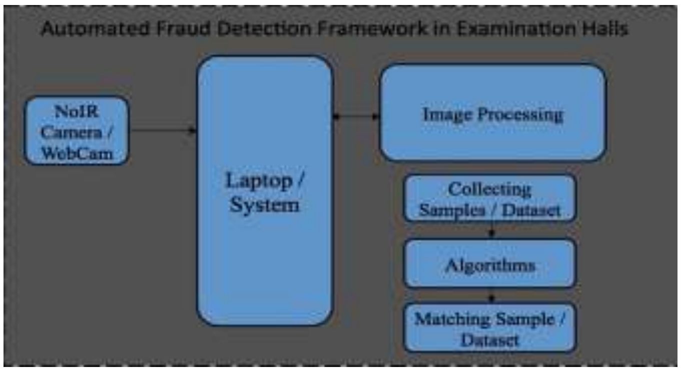

Automated Fraud Detection Framework in Examination Halls
NoIR Camera / WebCam
Laptop / System
Image Processing
Collecting Samples / Dataset
Algorithms
Matching Sample / Dataset

Fig -2: Proposed System Overview

The algorithms used in the proposed system include Random Forest Algorithm and Histogram of Oriented Gradients Algorithm for dataset training, recognition and classification.

3.1. Random Forest Algorithm

The random forest is a model made up of many decision trees. Rather than just simply averaging the prediction of trees (which we could call a "forest"), this model uses two key concepts that gives it the name random:

3.1.1. Random Sampling of Training Observations

When training, each tree in a random forest learns from a random sample of the data points. The samples are drawn with replacement, known as bootstrapping, which means that some samples will be used multiple times in a single tree. The idea is that by training each tree on different samples, although each tree might have high variance with respect to a particular set of the training data, overall, the entire forest will have lower variance but not at the cost of increasing the bias.

At test time, predictions are made by averaging the predictions of each decision tree. This procedure of training each individual learner on different bootstrapped subsets of the data and then averaging the predictions is known as bagging, short for bootstrap aggregating.

3.1.2. Random Subsets of Features for Splitting Nodes

The other main concept in the random forest is that only a subset of all the features are considered for splitting each node in each decision tree. Generally this is set to sqrt(n_features) for classification meaning that if there are 16 features, at each node in each tree, only 4 random features will be considered for splitting the node.

The following function can be written using random forest algorithm for dataset creation.

```
face_cascade = cv2.CascadeClassifier(haar_file)

webcam = cv2.VideoCapture(0)

count = 1

while count < 100:

(_, im) = webcam.read()

gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, 1.3, 4)

for (x,y,w,h) in faces:

 cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)

face = gray[y:y + h, x:x + w]

face_resize = cv2.resize(face, (width, height))

cv2.imwrite('%s/%s.png' % (path,count), face_resize)

count += 1

cv2.imshow('OpenCV', im)
```

The following function is written for dataset recognition.

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces:

 cv2.rectangle(im,(x,y),(x+w,y+h),(255,0,0),2)

face = gray[y:y + h, x:x + w]

face_resize = cv2.resize(face, (width, height))

prediction = model.predict(face_resize)

cv2.rectangle(im, (x, y), (x + w, y + h), (0, 255, 0), 3)

if prediction[1]<500:

cv2.putText(im,'%s-%.0f'%
(names[prediction[0]],prediction[1]),(x-10,y-10),
cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))

else:

cv2.putText(im,'not recognized',(x-10, y-10),
cv2.FONT_HERSHEY_PLAIN,1,(0, 255, 0))
```

3.2.Histogram of Oriented Gradients

The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

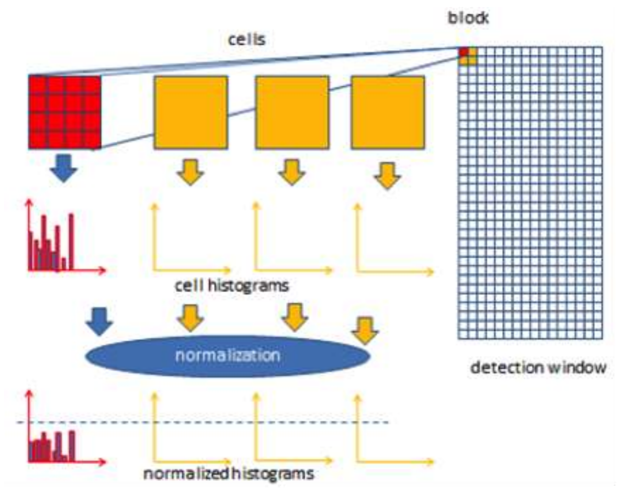


Fig -3: Overview of HoG Algorithm

The following function can be used for human detection in Histogram of oriented gradients algorithm.

```
# import the necessary packages from __future__ import print_function from imutils.object_detection import non_max_suppression from imutils import paths import numpy as np import imutils import cv2

cap = cv2.VideoCapture(0)

# initialize the HOG descriptor/person detector

hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

# loop over the image paths while True:

# load the image, resize it and improve detection accuracy
ret,image = cap.read() #image = imutils.resize(image, width=min(400, image.shape[1]))

# detect people in the image

(rects, weights) = hog.detectMultiScale(image, winStride=(4, 4), padding=(8, 8), scale=1.05)

# draw the original bounding boxes

for (x, y, w, h) in rects: cv2.rectangle(orig, (x, y), (x + w, y + h), (0, 0, 255), 2)

# apply non-maxima suppression to the bounding boxes using a fairly large overlap threshold

rects = np.array([[x, y, x + w, y + h] for (x, y, w, h) in rects])
pick = non_max_suppression(rects, probs=None,
```

overlapThresh=0.65)

# draw the final bounding boxes

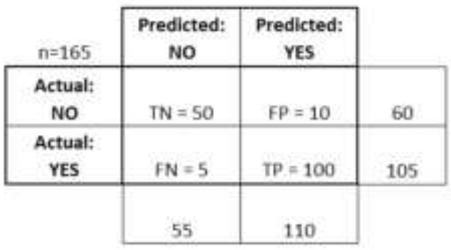
for (xA, yA, xB, yB) in pick: cv2.rectangle(image, (xA, yA), (xB, yB), (0, 255, 0), 2)

4. EXISTING VS PROPOSED MODEL

In the existing models, methods like ANN Classifiers and LBP method have been used. At a coarse scale, hall ticket identification system has been done in five phases: Preprocessing, Segmentation, Face detection, Feature extraction and ANN classification. The proposed system uses a low-cost camera to acquire snapshots of the scene. The images are fed into a feed-forward neural network, trained to detect the required anomalies. We aim to provide a higher accuracy than the existing system using the relatively more efficient HOG algorithm.

5. ACCURACY

The accuracy of the system has been measured on a small scale using a confusion matrix. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. Out of the 165 test cases provided 150 cases turned out to be successful.



n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Chart-1: Confusion Matrix

6. ADVANTAGES

- The predictive performance can compete with the best supervised learning algorithms
- They offer efficient estimates of the test error without incurring the cost of repeated model training associated with cross-validation.
- The HoG algorithm is indifferent to geometric and photometric transformations. Hence it is more suitable for human detection.

7. CONCLUSIONS

The proposed automated fraud detection framework in examination halls based on machine learning is thus a work in progress and a huge number of features can be added to make it more user friendly and efficient. Some of the features in development include adding an alarm system and candidate emotion detector system to detect the level of difficulty of the examination. The system will serve as a cost and performance efficient solution to monitor the behavior of candidates in an examination hall.

REFERENCES

[1] **IMAGE BASED FRAUD PREVENTION** (D.Madhu Babu, M. Bhagyasri, K.Lahari, C.H. Madhuri, G. Pushpa kumari Associate professer, Student Department of CSE, Lendi Institute of Engineering and Technology Vizianagram, India).

[2] **FACE RECOGNITION AND ITS APPLICATIONS** (Kumar, Naveen, and Sargam Badyal. "A Comprehensive Study of Drawbacks of Existing Face Recognition Techniques and Suggesting Improved Methods of Face Recognition.")

[3] **A NEW TECHNIQUE FOR LBP METHOD TO IMPROVE FACE RECOGNITION** ((Bouchaffra, Djamel. "Nonlinear topological component analysis: application to age-invariant faces recognition." Neural Networks and Learning Systems, IEEE Transactions on 26.7 (2015): 1375-1387.

[4] **AN AUTOMATED FRAUD DETECTION OF HALL TICKET IN AN OFFLINE EXAMINATION SYSTEM USING ANN CLASSIFIER**(Shridevi Soma, Vidhya Shree S.A. International Journal of Computer Applications Volume 126 – No.8, September 2015)

[5] **FRAUD DETECTION IN EXAMINATION USING LBP METHOD** (Tejashwini.S.G. International Journal of Latest Engineering and Management Research (IJLEMR)ISSN: 2455-4847