Intelligent Attendance and Database Integration Using Facial Recognition

Haroon ilyas[1]

Abstract

Face recognition technology is used in practically every industry in the modern world. Face recognition? It's everywhere now, and honestly, schools are jumping on that bandwagon too. Instead of wasting time with some teacher squinting at a list, calling out names like it's 1995, this new system, let's just call it the "Face Check Thing," does all the hard work. You stroll into class, the camera grabs your mug, and boom, attendance sorted. No more barcode scanning, no more "Oops, I forgot my ID." None of that. Here's the deal: every student gets their face snapped during sign-up, and that pic sits in a database somewhere safe (well, as safe as you can hope in 2024). When class starts, the camera just scans the room, matches faces with the database, and logs who's actually there. It's all automatic, so teachers can finally stop playing secretary and start, you know, teaching. Plus, the system cranks out attendance reports for the class, for the whole school, whatever you need. So, schools save time, cut down on paperwork, and probably spend less on those cheap plastic ID cards nobody remembers to bring anyway. Streamlined, efficient, and just a little bit Big Brother, but hey, welcome to the future.

Keywords: *Python, Image Processing, OpenCV, Face Detection, Face Recognition, Attendance System.*

Introduction:

Look, taking attendance is a pain; everyone knows it. Teachers waste a chunk of class time calling out names, and let's be honest, there's always that one sneaky kid answering for his absent buddy. Enter this facial recognition attendance system, built with Python and OpenCV and all that nerdy image processing magic. Basically, it snaps a pic of everyone in the room, runs it through a face recognition algorithm, and boom, attendance sorted, no roll call circus required (Shaik et al., 2024).

Teachers? They get to actually, you know, teach. No more "John? …John? …Is John here?" nonsense. Students win too, just show up, let the system do its thing, and check your attendance whenever you want (without needing to mess around in

[1] Scholar, University of Central Punjab, Lahore. Email: L1F20BSCS0883@ucp.edu.pk

https://doi.org/10.58932/MULL0006

the admin panel).

There's a proper structure to it: admins run the database, add new faces, and keep things tidy; lecturers pick when to snap the pics; students just have to bring their face, not their acting chops. And hey, if you're an admin, you're basically the Attendance Overlord, uploading, editing, managing, all that good stuff. No more paper sheets, no more fake signatures, just tech doing what it's supposed to: making life easier for everyone.

Figure 1:

Flow chart for Integrated Student Database and Attendance Management System with Face Recognition

The following are the functions of the lecturer: Start the students' attendance; View the attendance; Retrieve the queries; Control over the system; Manually mark the students' attendance; Time.

Related Work:

Let's make this sound like an actual person wrote it, maybe someone running on too much coffee and not enough sleep: So, Hajar Philadelphia (Riffi et al., 2018) (no, not the city, apparently a person) and Aluminum seriously, that's the name? took a look at four machine learning tricks: Haar-AdaBoost, LBP-AdaBoost, GF-SVM, and GFNN. Basically, they wanted computers to stop acting dumb and start learning to handle tough stuff, instead of just relying on old-school algorithms (Toppo & Tracy, 2021).

The first two, Haar-AdaBoost and LBP-AdaBoost, are all about that boost life. They use the boosting algorithm to pick out and train what they call "strong classifiers (Cao et al., 2013)." Think of it like a talent show, but for computer programs, with only the best cutting. Waterfall classification is the name of the game here, whatever that means (it sounds cool though) (Verenych & Semenov, 2023).

The last two GF-SVM and GF-NN throw Gabor filters into the mix. These filters are like the computer's way of squinting at an image to pick out the juicy details, which supposedly helps with classification (Recio et al., 2005). Anyway, after messing around with all these methods, the big takeaway is: the choice of approach actually makes a real difference, especially when it comes to how quickly stuff gets detected.

Out of the four, the AdaBoost hair method is still the speed demon. So yeah, we're sticking with that one. Oh, and those folks in Ahmed et al. (2018)? They came up with a way to dodge the pain of the old-fashioned, "sign your name and hope nobody cheats" attendance system. This article explains how you can use real-time face recognition and detection to confirm student attendance. The article illustrates a camera set in the classroom that collects photographs and can identify several faces as part of an automatic attendance system. This method has various phases, such as establishing a student face database, using HOG functions, recognizing faces and eyes, using an SVM classifier, comparing faces, and determining attendance (Kapse et al., 2022). The SVM classifier, the Viola-Jones and HOG functions, and other algorithms are employed to produce the desired outcomes (Xu et al., 2016). The system could be light-sensitive because of flaws in the card. This flaw can be fixed in the suggested system by utilizing algorithms that are not sensitive to illumination and using advanced high-resolution cameras.

Research Methodology

OpenCV:

For the analysis of photos and movies, the well-known open-source computer vision package OpenCV offers a number of tools and techniques. It is a well-liked framework for creating face recognition applications since it has built-in capabilities for face detection and recognition. For both face detection and face identification, OpenCV offers powerful and effective algorithms. The OpenCV video and image processing framework is used for a number of applications in image and video analysis, including advanced robotic vision, facial recognition, license plate reading, and photo retouching.

The most effective and well-known computer vision library is OpenCV. It supports a variety of languages, including Python, C/C++, Java, and others.

We printed the image in a matrix format in SimpleCV. The imshow() method will now be used to output the actual image with the aid of OpenCV.

```
import cv2
img=cv2.i
mread("co
mp.jpg",1)

cv2.imshow("source",resize)
resize=cv2.resize(img,(int(img.shape[1]/4),int(img.shape[0]/4))

cv2.waitKey(0)

cv2.destroyAllWindows()
```

The photograph will be displayed by the aforementioned programme at a 4x reduction in size. Use the Imshow() function to display your image. The image is shown for the number of milliseconds set by the waitKey() method. Any key you press after writing 0 will close that.

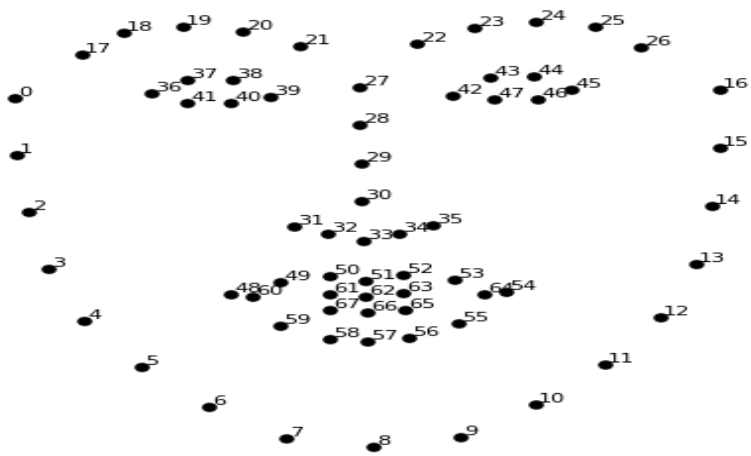
Window. If you enter 25, the window will remain visible for 25 milliseconds before being destroyed with the destroyAllWindows() method.

Face Detection:

Our workflow's initial step is face detection. Naturally, identifying faces in a photo is important before making an effort to differentiate between them. Face detection is a terrific feature for cameras. The camera may make sure that every face is in focus before shooting the picture if it has the ability to recognize faces automatically. So, here's the deal: another big reason for using this thing is to carve out the exact parts of the image we want to toss down the line in our pipeline. Flashback to the early 2000s: Paul Viola and Michael Jones dropped a face detection technique that was actually fast enough to work on cheap cameras. Kind of wild at the time, trust me. But nowadays? Way more solid options out there. What we're rolling with is called Histogram of Oriented Gradients (or just HOG if you're into the whole brevity thing). Came out in 2005. Basically, HOG zeroes in on the shapes and outlines of objects. It's slick because it doesn't just care about how strong the edges are; it also cares which way they're pointing. That combo makes it better than the old-school edge detectors.

Figure 2:

Landmark on Face: ***There are 68 landmarks we will locate on every face.***

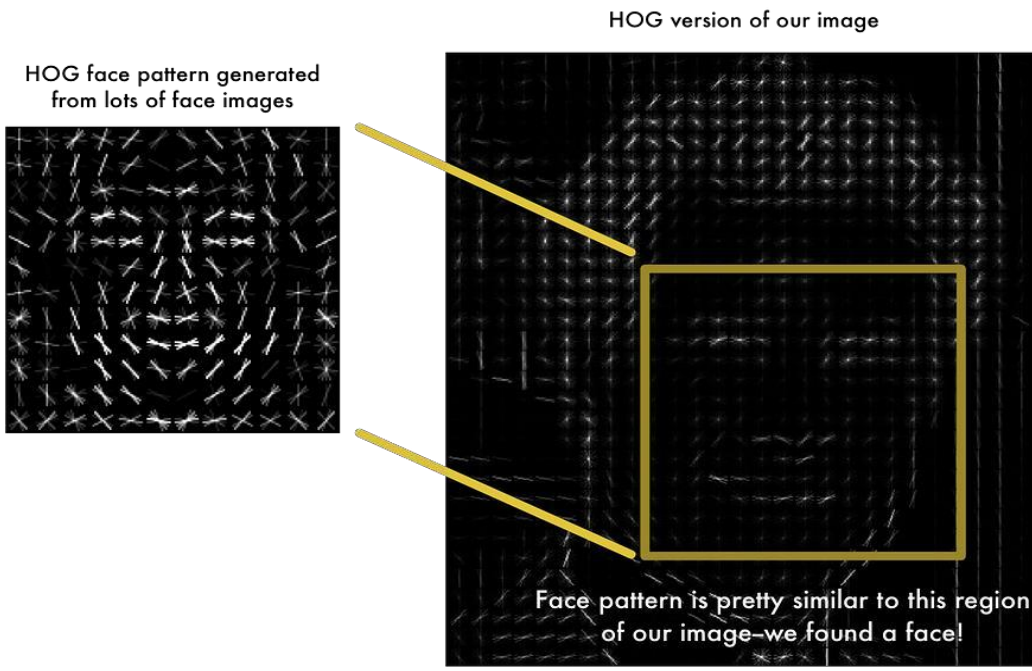


- Training Data Gathering: A collection of positive samples containing images and negative samples, faceless images is gathered. The face regions are often indicated by bounding boxes on the positive samples.
- Feature Extraction: The HOG descriptor, which records the shape and gradient information, is calculated for each training sample. In order to do this, gradient magnitudes and orientations within discrete image cells must be calculated, and histograms must be created using these results.

Classifier Training: Based on the HOG features, a binary classifier, such as Support Vector Machines (SVM) or AdaBoost, is trained to distinguish between face and non-facial regions.

Figure 3:

The Original Image Is Turned Into A HOG Representation That Captures The Major Features Of The Image Regardless Of Image Brightness.



Face Recognition:

Alright, so here's the deal with face recognition. Once the system spots a face in a photo, it lines it up against a stash of faces sitting in its database. If you're using OpenCV? Honestly, it's pretty solid, like, you'll get a hit at least 9 out of 10 times, assuming the image isn't potato quality.

Now, if you're rolling with HOG (that's Histogram of Oriented Gradients, but who actually says the full thing?), Here's the usual playbook:

First, you've got to make sure the face is, you know, facing the right way. So, the software tweaks and normalizes the mugshot kind of like making everyone look straight ahead for a driver's license photo. This helps cut down on weird angles and makes it way easier to pull out the good stuff.

Next move: feature extraction. This is where HOG comes in. It snags essential info about the face edges, lines, and all those little gradient details that make you look like you. Basically, it's looking for facial "fingerprints," except less CSI.

Then, on to training. The system grabs all those HOG features and feeds them to a machine learning model, usually something like SVM or k-NN. You load it up with a bunch of labeled faces (think: "this is Bob, that's Alice, here's some random from accounting"), and the model learns who's who.

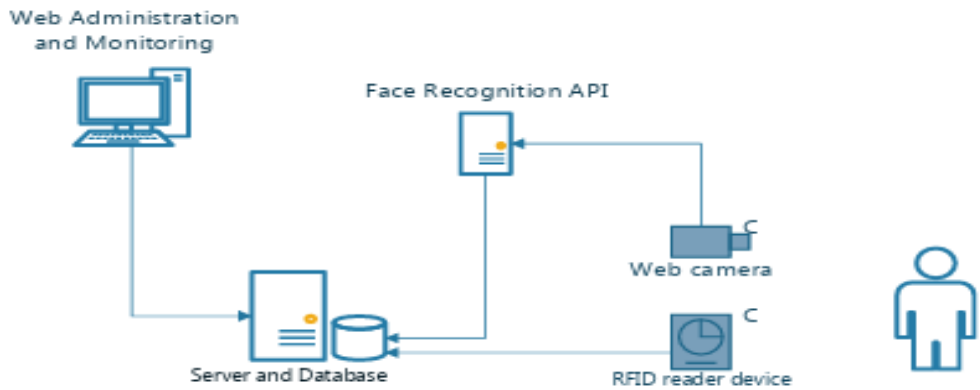
Finally, recognition time. The system breaks down the new face with HOG again, tosses those features into the trained model, and boom, it spits out who it thinks you are. Sometimes it'll just tell you the best match, other times it'll give you a confidence score, like, "Hey, I'm 92% sure this guy's Steve." Not bad, right?

Attendance System:

An attendance system is a technical solution that allows businesses and institutions to track and manage the attendance of their workforce (staff or students, or members). Typically, it involves automating the process of tracking and recording attendance through the use of various technologies, including biometrics and mobile applications. Real-time Attendance tracking: Attendance systems may offer real-time attendance tracking, enabling administrators or supervisors to see a person's attendance status at any given time. With the use of this function, absentees or tardy attendees can be quickly identified and dealt with. The faces that were recognized by facial recognition will therefore be marked as present on the Excel sheet, whereas the faces that were not recognized will be marked as absent. The appropriate faculties will thereafter get a letter with the list of absentees. A revised monthly attendance form will be given to faculty at the end of each month.

Figure 4:

Working Mechanism of the Attendance System



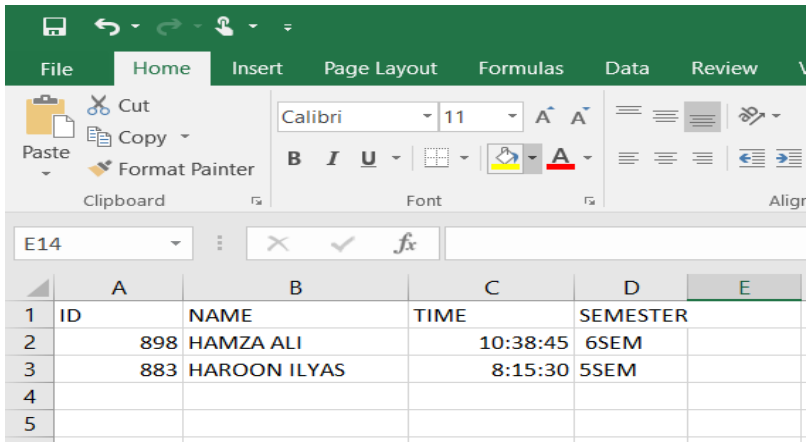
When the camera starts on its own. The facial identification pane in Fig. 1 shows recognized faces; if they had not been recorded, it would have displayed "unknown". You can close the window, update the attendance data in the Excel spreadsheet, and mail the names of the students who were absent to the proper faculty by pressing CTRL+Q. Following the recognition procedure, the revised attendance sheet is depicted in Fig. 2.

Figure 5:

Face Recognition

Figure 6:

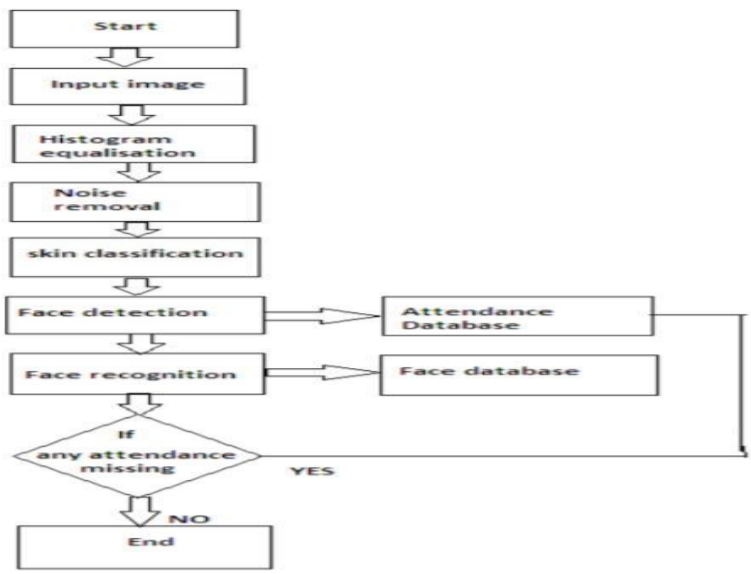
Attendance Sheet



	A	B	C	D	E
1	ID	NAME	TIME	SEMESTER	
2	898	HAMZA ALI	10:38:45	6SEM	
3	883	HAROON ILYAS	8:15:30	5SEM	
4					
5					

Figure 7:

Flow chart for Integrated Student Database and Attendance Management System with Face Recognition

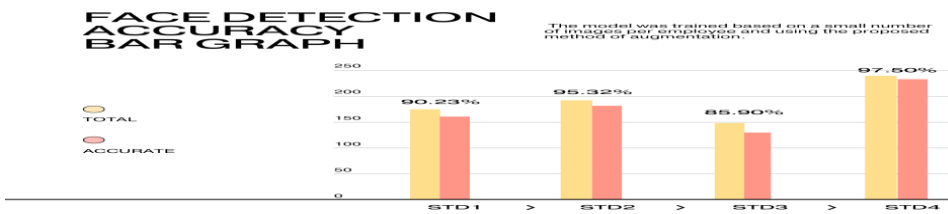


Results and Discussion: • Face detection, detail wrangling, and dataset training: So, here's the deal first, you've got to get the students in the system. There's

this section where you plug in all their info (think: name, ID, the usual suspects). For the face recognition part, it's not just one awkward passport photo. Nope, you've got to snap like 10 different photos of each student. Why? The system needs a bunch of angles and expressions to really "get" their face. Not exactly glamorous, but hey, it works. The students' photos that were taken are kept in a local database. A classifier is used to train the stored photos, and related labels like ID, name, and department are assigned.

Figure 8:

Accuracy OF Face Detection Per Class Is Presented In The Above Figure. The Overall Accuracy Of The System Is 92.23%.

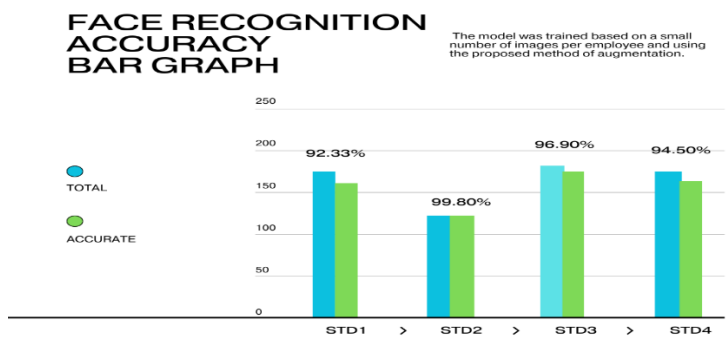


Face Recognition and Attendance Result stored in an Excel sheet:

The features that are compared to the features that are kept in the database are those used for face recognition. Following face recognition using the defined datasets, the student's face is displayed together with the accompanying ID, Name, and Department, which are used to record attendance.

Figure 91:

The Accuracy Of Face Recognition Per Class Is Presented In The Above Figure. The Overall Accuracy Of The System Is 95.88%.



An Excel sheet contains the pupils' corresponding attendance data. The data can be exported and imported using the buttons provided in the attendance menu. The file is saved as a .csv file.

Figure 10:

An Excel Sheet Contains The Pupils' Corresponding Attendance Data.

ID	NAME	TIME	SEMESTER
898	HAMZA ALI	10:38:45	6SEM
883	HAROON ILYAS	8:15:30	5SEM

Conclusion:

Look, face recognition attendance systems? Total game-changer for organizations trying to keep tabs on who's actually showing up. No more buddy punching or sneaky "sign in for me" nonsense; these things are way too smart for that. Just scan your mug, boom, you're logged. Kind wild, honestly. Biometrics aren't just about convenience either. They crank up security a notch and keep those attendance records squeaky clean. Plus, with all that real-time data flying around, it's not just about who's here or not the big bosses can spot patterns, fix scheduling headaches, and make decisions based on facts instead of gut feelings. Bottom line? If you roll these systems out the right way, you get sharper accuracy, less paperwork, and a whole lot less room for error. It's like giving your admin team a turbo boost. Just don't skimp on setup, or all that fancy tech will be wasted. Oh, and about sorting algorithms (yeah, bit of a nerdy tangent, but stick with me), future research needs to dig into how these things really perform out in the wild. Try different algorithms, mess with optimizations, see how they run in various programming languages, all that jazz. The more we know about their quirks, the better we can make them. This whole study? It's not just academic hand-waving. The numbers actually matter, and they're a solid jumping-off point for folks who want to push sorting algorithms to the next level. So yeah, let's keep poking at this stuff and see what shakes loose.

References:

Ahmed, A., Guo, J., Ali, F., Deeba, F., & Ahmed, A. (2018). LBPH-based improved face recognition at low resolution. In Proceedings of the International Conference on Artificial Intelligence and Big Data (pp. 1–6).

Cao, Y., Miao, Q.-G., Liu, J.-C., & Gao, L. (2013). Advances and prospects of AdaBoost algorithm. Acta Automatica Sinica, 39(6), 745–758. https://doi.org/10.1016/S1874-1029(13)60052-X

Cognotics. (2007). Seeing with OpenCV. http://www.cognotics.com/opencv/servo_2007_series/part_1/index.html

Emami, S. (2010). Face detection and recognition using OpenCV. http://shervinemami.info/faceRecognition.html

Kapse, A., Kamble, T., Lohar, A., Chaudhari, S., & Puri, D. (2022). Face recognition attendance system using HOG and CNN algorithm. ITM Web of Conferences, 53, Article 03028. EDP Sciences. https://doi.org/10.1051/itmconf/20225303028

Recio, J., Fernández, L., & Fernández, A. (2005). Use of Gabor filters for texture classification of digital images. Física de la Tierra, 17, 47–59.

Riffi, H. F. J., Mohamed, A., Mahraz, M., & Tairi, H. (2018). Multiple face detection based on machine learning. In Proceedings of the IEEE International Conference (pp. 1–6).

Shaik, V. M. S., Kumar, M., Praveen, V., & Potti, S. (2024, May). Facial recognition based attendance system using OpenCV. Publication Hub.

Toppo, G., & Tracy, J. (2021). Running with robots: The American high school's third century. MIT Press.

Verenych, O., & Semenov, M. (2023). Agile and waterfall approaches hybridization in hyper-casual games projects. In 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS) (pp. 477–480). IEEE. https://doi.org/10.1109/IDAACS58523.2023.10348642

Xu, Y., Yu, G., Wang, Y., Wu, X., & Ma, Y. (2016). A hybrid vehicle detection method based on Viola–Jones and HOG+SVM from UAV images. Sensors, 16(8), 1325. https://doi.org/10.3390/s16081325