IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

KumbhSetu: An Integrated Emergency Command, Crowd Safety & Pilgrim Assistance Platform for Nashik Kumbh Mela

Kaveri Yogesh Deshmukh[1], Punam Kailas Gaikwad[2], Sakshi Sanjay Bhad[3], Sakshi Madhuka Pabale[4], Abhale B. A[5], Khalse R. A[6]
Department of Information Technology
S. N. D. College of Engineering & Research Center, Yeola, Nashik
kaveriyd11@gmail.com, gaikwadpunam2255@gmail.com,
sakshibhad429@gmail.com, sakshipabale32@gmail.com

Abstract: *Kumbh Mela brings millions of pilgrims into a small area, so a packed lane, a small fire or a person who has fallen can turn into a disaster within minutes, and control-room staff cannot watch every corner by eye. This paper first reviews seven sources on crowd management, crowd counting, fire detection and fall detection, and then presents the design of a web-based system for the Nashik Kumbh Mela. The system studies uploaded video, and live streams if speed and accuracy allow, to count people zone by zone, spot fire and notice falls. The results appear on a dashboard used by a super admin, who can approve volunteers, hand out tasks and follow their progress. Volunteers log in to see their tasks and mark each one as in progress, completed or incomplete, while pilgrims can register, open a map of emergency, medical and other key places and send an SOS or help request. The system is planned with Python, Django, OpenCV and deep-learning models, with Firebase or SQLite for data and Bootstrap for the screens. The review shows that earlier work solves one part of the problem and that none joins monitoring, volunteer work and pilgrim help in one place. The paper also explains the architecture, the working steps, the test plan and the limits of the design.*

Keywords: Kumbh Mela, crowd management, crowd counting, fire detection, fall detection, Django, volunteer coordination, public safety

I. INTRODUCTION

The Kumbh Mela ranks among the biggest gatherings of people anywhere on Earth. Press reports on the 2025 Maha Kumbh at Prayagraj spoke of roughly 50 to 60 lakh visitors every day, and of several crore on the principal bathing days [1]. It is a moving sight, yet it is also extremely hard to manage. On a narrow bridge, in a blocked lane or beside a small fire, a calm scene can turn dangerous within a few minutes. Many people still remember the 2013 Kumbh at Allahabad, where a stampede on a bridge claimed 36 lives [2].

Most control rooms still depend on operators who watch rows of screens. That is fine for a handful of cameras and it breaks down when there are hundreds. Attention fades after long shifts, and it is easy to overlook a crowded corner, a flame at the edge of the frame or a person lying on the
ground. For the 2025 event the authorities put up about 1,800 cameras across the fair area, but only around 160 of them carried AI features [1]. The rest of the footage still needs a human to notice trouble.

People on the ground need help as well. A pilgrim wants to know where the nearest hospital is, which road is jammed and how to call for assistance without losing time. Volunteers need clear instructions about where to go and what to do. If there is no shared task list, a dozen of them may cluster around one gate while another gate has nobody at all.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

The place itself adds to these worries. The Nashik Kumbh is centred on the Godavari, and the Ramkund area in particular draws heavy crowds into a compact stretch of lanes and ghats. Tools made for wide, open grounds do not always suit such a layout, so a design for Nashik is better planned zone by zone, with a separate limit and a separate alert for each place.

This paper has two aims. The first is to review seven recent and widely known sources [1]–[7] and to point out what each of them leaves uncovered. The second is to propose a web-based system for the Nashik Kumbh Mela that brings crowd counting, fire detection, fall detection, volunteer management and pilgrim assistance together. The system is written in Python on the Django framework [8], so a single platform can serve the super admin, the volunteers and the pilgrims.

In short, the paper contributes a structured review of the literature, a comparison table showing which features each study offers, a complete architecture for the proposed system and a test plan with measurable targets that a later pilot can check. No claim of measured accuracy is made, because the prototype has not yet been run on real Kumbh footage.

The remaining sections are organised as follows. Section II reviews the related work and the gaps it leaves, Section III describes the proposed system in detail, Section IV presents the test plan, the expected outcomes and the limits of the design, and Section V closes the paper with conclusions and future work.

II. LITERATURE REVIEW

Work that matters for safety at Kumbh scale can be sorted into four groups: crowd management at the level of a whole event, crowd counting from images, fire detection and fall detection. The seven sources are reviewed under these four headings, and the section ends with a summary of the gaps that remain.

A. Crowd Management at Kumbh Scale

The report on the 2025 Maha Kumbh [1] describes AI-enabled cameras that estimate how many people occupy each square metre of a selected spot. Officials used these figures to predict how many pilgrims would turn up on the big bathing days and to divert people before an area became too full. Monitoring was run from four command centres inside the fair area. The report describes this as the first use of AI for crowd control on such a scale, although it does not publish any accuracy figures.

Choudhury et al. [2] put forward Drishti AI-Event Guardian, a research framework meant for mass events. It takes video from fixed cameras and drones, estimates crowd density with a YOLOv8 model, flags unusual movement and predicts how a crowd is likely to flow. It can also look for missing persons through face recognition, accept medical reports and send guards towards crowded zones. On a Kumbh Mela scene and a parade scene the authors report a density error of 3.2 persons per square metre, an F1-score of 0.91 for unusual-movement detection and a median alert delay of 111 ms.

Both works show that AI can be useful at a Kumbh, and both also leave room for improvement. The report in [1] offers no measured results. The framework in [2] runs on a cloud AI platform, and the modules its authors list do not include fire detection, fall detection or a step in which volunteers are approved and given tasks.

One point from these works is worth carrying forward. A density figure in persons per square metre is more useful to an officer than a raw head count, because the same number of people can be safe in an open field and risky in a narrow lane. This is one reason why the proposed system sets the limit separately for every zone and does not use a single limit for the whole fair.

B. Crowd Counting Methods

Counting people in a packed image is hard because heads look tiny, overlap one another and change size with distance from the camera. Zhang et al. [3] deal with this through a network of several columns, each tuned to heads of a different size. Li et al. [4] built CSRNet, which keeps the front end of a standard VGG-16 and replaces pooling in the back end with dilated convolutions. In this way fine detail is preserved while a wide area is still seen. The network

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

draws a density map, and the sum over the map gives the head count. The authors report that on the ShanghaiTech Part B set the counting error fell by 47.3% compared with the best earlier method.

YOLO [5] takes a different route. It finds every object in one pass over the image and is fast enough for real-time use, which suits scenes where people can still be told apart one by one. The limit shared by all three studies is that they are model studies. They do not explain how a count should reach a control room, a volunteer or a pilgrim.

Counting studies usually report two error measures, the mean absolute error (MAE) and the mean squared error (MSE), computed over a set of test images [3], [4]. MAE shows how far the predicted count is from the true count on average, while MSE gives more weight to large misses and so says something about how stable a method is. Both measures are reused in the test plan of this paper. A further choice lies between detection and density regression. Detection draws a box around each person and works well when people are separate, whereas density regression does not need every head to be located and copes better when the crowd is packed. A practical system may need both.

C. Fire Detection from Camera Video

Abdusalomov et al. [6] adapted YOLOv3 to find fire in surveillance video, by day and by night. They trained it on a large set of fire pictures and enlarged the set by rotating each picture and changing its brightness. They also added fire-like pictures, such as neon signs, street lights and vehicle headlights, because these often fool detectors. The authors report that the method could spot a fire about 1 m long and 0.3 m wide from a distance of 50 m, and that it ran on a low-cost board.

Two lessons matter for a Kumbh. Fire can be found in ordinary camera video without special heat or smoke sensors, and bright lights have to be handled with care to avoid false alarms. Lamps and decorated stalls are found everywhere at a mela, especially after dark.

A second practical point concerns the alert rule. A detector scores every frame on its own, and one frame is a weak basis for sounding an alarm, since a passing flash or a reflection can look like flame for an instant. For this reason the proposed system waits for the same finding in several frames in a row before it raises an alert, and it leaves the final decision to a person who looks at the saved snapshot.

D. Fall Detection from Camera Video

Gutiérrez et al. [7] reviewed vision-based fall detection and found that the field has grown quickly. The systems they studied follow, with small differences, a common three-step approach that begins with preparing the video signal. Camera-based methods need no device on the body, which counts for a lot at a mela, where pilgrims cannot be asked to wear anything. A person who falls inside a crowd risks being trampled, and tired or elderly pilgrims are at higher risk, so a fall module deserves a place in a Kumbh system.

Falls are difficult to study because they are rare, they are over in a moment and they cannot be staged freely without risk to the actors. Public sets such as Le2i and URFD contain acted falls recorded mostly indoors, and a pilgrim falling in a crowded outdoor lane looks quite different. For this reason the fall module in this paper is treated as an early-warning aid that must be tested on footage resembling the real site, and its first use is planned in lanes, gates and medical areas where people are not packed shoulder to shoulder.

E. Summary of Gaps

Table I places the six technical studies beside the proposed system. A “No” means that the feature is not reported in the cited work, “Part” means that it is mentioned only in part, and “Planned” means that the proposed system intends to offer it but must first prove it through testing. Three patterns stand out. First, each study covers a single slice of the problem, and none of them combines counting, fire detection and fall detection on one platform. Second, only [2] touches the human side of an emergency, and none of the sources gives pilgrims a map or a way to ask for help. Third, the results come from test sets and model trials, not from a complete system that staff, volunteers and visitors use together.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

From this review, five needs follow for the proposed system. It should work zone by zone, because dangers differ from place to place. It should cover more than one danger, since crowding, fire and falls can occur on the same day at the same site. It should keep a person in the loop, because every detector makes mistakes. It should connect a detection to an action, which means tasks for volunteers and a way for pilgrims to call for help. Finally, it should run on tools that a state or city office can afford and maintain, which points towards open-source software and ordinary hardware. These five needs guide the design in the next section.

TABLE I: COMPARISON OF REVIEWED STUDIES WITH THE PROPOSED SYSTEM

Feature	[2]	[3]	[4]	[5]	[6]	[7]	Proposed
Crowd counting	Yes	Yes	Yes	No	No	No	Yes
Fire detection	No	No	No	No	Yes	No	Yes
Fall detection	No	No	No	No	No	Yes	Yes
Emergency / SOS	Yes	No	No	No	Part	No	Yes
Volunteer tasks	Part	No	No	No	No	No	Yes
Pilgrim map and help	Part	No	No	No	No	No	Yes
Admin dashboard	Part	No	No	No	No	No	Yes
Live video input	Yes	No	No	Yes	Yes	Part	Planned
Measured results	Yes	Yes	Yes	Yes	Yes	Part	Planned

Reading Table I by columns, no single study ticks more than three of the first seven rows, while the proposed system aims to tick all of them. Reading it by rows, the weakest areas are volunteer tasks, the pilgrim map and the admin dashboard, which are the features that decide whether a detection actually leads to help on the ground. The last two rows also remind the reader that the proposed system is still a design: live video and measured results are marked "Planned" and not "Yes", because they must first be earned through testing.

III. PROPOSED METHODOLOGY

The proposed system is a web platform that watches video, finds problems and passes the news to the right people. Fig. 1 shows the block diagram. Video from an uploaded file or a camera goes to the image-processing and AI unit, which counts people, looks for fire and looks for falls. The results move on to a Django backend that holds the rules of the system and talks to a database, either Firebase or SQLite. Three kinds of users work through the backend: the super admin, the volunteers and the pilgrims. Fig. 2 shows the working steps as a flowchart.

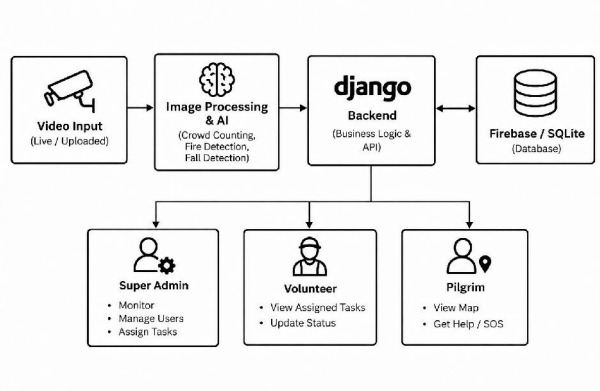


Fig. 1. Block diagram of the proposed Nashik Kumbh Mela crowd monitoring and assistance system.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

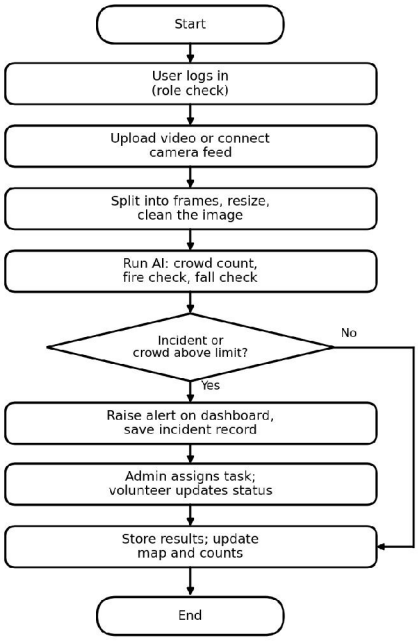


Fig. 2. Operating flowchart of the proposed system.

A. Video Input and Pre-Processing

The first version works on uploaded video files, since they are easy to test and need less computing power. Live camera streams will be added only if the system reaches enough speed and accuracy on the available hardware. OpenCV reads the video and takes one frame every second or two instead of every frame, which saves time without losing much. Each frame is resized and its brightness is corrected. Every video or camera is tagged with a zone name, such as the Ramkund area, so that all results can be shown area by area.

Skipping frames is a deliberate choice. A person who falls or a flame that starts does not vanish in a second, so one frame every second or two is enough for most events, and it leaves room on ordinary hardware to run three modules together. The sampling interval can be set by the admin for each zone. A busy gate may be sampled more often than a quiet field, and a camera that is switched off at night can simply be skipped.

B. Crowd Counting Module

The counting module uses a density-map network in the style of CSRNet [4], trained on public crowd sets such as ShanghaiTech and then checked on local footage. Where the crowd is thin and people can be told apart, a YOLO person detector [5] can be used in its place. The count of each zone is averaged over a few recent frames so that the number does not jump up and down. The super admin sets a limit for each zone. A zone is shown as low, medium or high in green, orange or red, and a warning is raised when the count crosses its limit.

The limits are chosen by the admin and not fixed in the code, since a safe count for a wide ghat is very different from a safe count for a lane near a temple. The three colour bands help a tired operator read the dashboard at a glance. Orange is meant as an early hint that a zone is filling up, and red is meant as a call to act, for example to send volunteers to slow down the entry of new people. If the counts of a zone rise steadily over several minutes, the dashboard marks the zone with an arrow so that the trend is visible before the red band is reached.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

C. Fire and Fall Detection Modules

The fire module uses a YOLO-type detector [5], [6] trained on fire and smoke pictures, together with many fire-like lights as negative examples. A fire alert is raised only when fire is seen in several frames in a row, which cuts down false alarms from flickering lamps. The fall module first finds each person and the main body points. It then watches for a sudden drop in body height, a body that turns flat and a person who stays still on the ground for a few seconds [7]. Staged fall videos, and public fall sets such as Le2i and URFD, will be used for training and testing. Every alert is saved with its time, zone, type and one snapshot.

Each alert carries a short priority tag. A fire alert is marked high, a fall alert is marked high when it comes from a lane or a gate and medium otherwise, and a crowd warning follows the colour band of the zone. The tag decides the order of the alert list, so that the most urgent item is always at the top. All three modules write to the same record format, which keeps the dashboard simple and makes it easy to add a fourth module later, for instance for smoke alone or for people moving against the main flow.

D. Django Backend and Database

Django handles login, user roles, forms, task records and the links through which the AI unit sends its results [8]. Each detection becomes one record that holds the time, zone, type, count, snapshot and status. SQLite is simple and quite enough for a single server and for development, while Firebase can push updates quickly to many phones and screens at once. The final choice will be made at deployment. Django stores passwords in hashed form and includes protection against cross-site request forgery. Every page also checks the role of the user, and the site should be served over HTTPS.

The AI unit and the web pages talk to each other through a small set of Django views. The AI unit posts a detection record, the dashboard asks for the latest records of each zone, the volunteer page asks for the tasks of one user and the pilgrim page posts an SOS. Keeping these links small and separate means that the AI unit can be moved to a stronger machine later without touching the screens.

E. User Roles and Screens

The super admin sees a dashboard with zone-wise counts, a list of fire and fall alerts and the incoming SOS requests. The admin can manage users, approve or reject volunteer sign-ups, assign tasks and follow their progress. A volunteer registers on the site and cannot log in until the admin approves the account. After that, the volunteer sees the tasks given to him or her and marks each one as in progress, completed or incomplete.

A pilgrim registers, logs in and opens a map of the Nashik Kumbh Mela area. The map marks emergency services, medical facilities and other key points, and the pilgrim can send a help request or an SOS to the admin. If the counting data proves reliable, the crowd level of each zone is shown on the map as well, so pilgrims can steer clear of packed areas. The screens are built with Bootstrap so that they fit a mobile phone, which is the device most pilgrims will carry. The technologies used are listed in Table II.

A volunteer needs very little to start working. After the admin gives the approval, the volunteer sees a short list that shows the place, the reason and the time of each task, and a single tap changes its status. The admin sees that change at once. If a task stays in progress for too long, the dashboard can highlight it so that the admin may send a second volunteer. In this way the task list works as the shared picture that the introduction found missing.

TABLE II: TECHNOLOGIES USED IN THE PROPOSED SYSTEM

Component	Function
Python	Main language for all logic
OpenCV	Reads video, makes frames
Deep-learning models	Count people, find fire and falls
Django	Backend, login, user roles, APIs
Firebase / SQLite	Stores users, tasks, alerts

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/568

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

Component	Function
Bootstrap	Web screens for phone and PC
Map library	Shows key places and crowd level
Browser / mobile	Used by admin, volunteers, pilgrims

F. Working Sequence

A user first logs in, and the system checks the role. The admin uploads a video or connects a camera for a zone. The system splits the video into frames, cleans them and runs the three AI modules. If the count is above the limit, or fire or a fall is found, an alert is shown on the dashboard and saved as a record. The admin checks the alert, assigns a task to a free volunteer and follows its status until it is closed. If nothing is wrong, the results are stored and the map and counts are refreshed. Pilgrims can send an SOS at any time, and it reaches the admin directly.

G. Database Design

The data is kept in a few simple tables. The user table holds the name, the contact details, the role and the approval status. The zone table holds the zone name, its limit and its map position. The detection table holds one row for every count, fire alert or fall alert, with its time, zone, type, value, snapshot and status. The task table links one volunteer, one detection or SOS and a status of in progress, completed or incomplete, and the SOS table holds the pilgrim, the message, the time and the state of the request. Links between the tables are kept through ordinary keys, so that a single alert can be followed from the first detection to the task that closed it. Such a trail is also useful later, when the officers want to learn how quickly each kind of trouble was handled.

H. Alert Handling and Task Assignment

When a record needs attention, the dashboard plays a sound and shows the alert with its snapshot, its zone and its priority tag. The admin may confirm the alert, mark it as a false alarm or turn it into a task. A confirmed alert is given to one volunteer or to several, and each of them receives it on the task page. A false alarm is not deleted. It is kept with a flag, because the number of false alarms is itself one of the measures in the test plan and is also useful for tuning the detectors. An SOS from a pilgrim follows the same path, except that it appears at the top of the list with the highest priority and keeps blinking until an admin has opened it.

I. Security and Privacy

Three groups of users mean three levels of access. A pilgrim can see only his or her own requests and the public map. A volunteer can see only the tasks given to him or her. Only the super admin can see counts, snapshots and the full list of users. These rules are checked on the server for every request and are not left to the screen alone. Passwords are hashed, forms carry the forgery token that Django provides, and the whole site should be served over HTTPS. On the privacy side, the system keeps counts and a few snapshots of incidents instead of full video, makes no attempt to recognise faces, and removes old snapshots after the period allowed by the authorities.

J. Implementation Plan

The work is planned in five steps. In the first step the video reader and the three detectors are built and tried on public data. In the second step the Django backend, the database and the login system with the three roles are made. In the third step the admin dashboard, the volunteer page and the pilgrim map are joined to the backend. In the fourth step the whole system is run on staged and recorded videos, and the tests of Table III are carried out. In the last step the weak points found by the tests are repaired, and the system is prepared for a small trial with the local authorities. Keeping the steps in this order means that a working core exists early and every later step can be checked against it.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 3, November 2026 Impact Factor: 8.2

K. Scenario Walkthrough

Three short scenarios show how the parts fit together. In the first, a camera above a lane near Ramkund shows the count of its zone rising past the orange band. The dashboard marks the zone, the admin sees the upward arrow, and a volunteer near the nearest gate is asked to slow the entry of new people. A few minutes later the count settles and the admin closes the task.

In the second scenario, a stall at the edge of the fair catches fire after dark. The fire module sees flame in several frames in a row, raises a high-priority alert and saves one snapshot. The admin looks at the snapshot, sees that it is a real flame and not a decorative lamp, and turns the alert into a task for the nearest volunteers, who can then call the responsible service. The saved record shows later how long each step took.

In the third scenario, an elderly pilgrim feels dizzy and sits down on the ground in a quiet lane, and a family member presses the SOS button on a phone. The request reaches the admin at the top of the list, the map shows the nearest medical point, and a volunteer is sent with the details. If the fall module has also noticed the person lying still, the two signals support each other, and the admin can act with more confidence. These examples are only illustrations of the intended use, and they will be turned into staged tests during the pilot.

IV. RESULTS AND DISCUSSION

A. Evaluation Approach

This paper presents the design and the test plan. The full system has not yet been tested on real Kumbh footage, so no accuracy figures are claimed here. This section explains what the design is expected to deliver and how it will be judged. Among the sources reviewed, [2]–[6] report numbers for their own models, but none tests a complete system that joins monitoring, volunteers and pilgrims. The planned tests are meant to fill that gap.

The first expected outcome is a dependable crowd picture. People will be counted by hand in a sample of frames from the test videos, and the count given by the system will be compared with the hand count. The second is quick and trustworthy fire and fall alerts. Short clips with and without fire will be played to the system, and both the correct alerts and the false alarms will be noted. For the fall module, clips with a real fall, sitting down, bending and lying down on purpose will be mixed, so that normal actions are not mistaken for falls. The third outcome is speed: the time from an event in the video to an alert on the dashboard, and from a pilgrim pressing SOS to the admin seeing it.

The fourth outcome is smooth teamwork between users. A volunteer account must stay locked until the admin approves it, a task must show the same status to the volunteer and to the admin, and the map must open and respond on an ordinary phone. The measures and their planned criteria are given in Table III. These criteria are engineering targets for a pilot and not measured results.

For the counting module the main measures will be the MAE and the MSE against hand counts. For fire and fall alerts the main measures will be precision, which tells how many of the raised alerts were real, and recall, which tells how many of the real events were caught. The F1-score joins the two into a single figure. Both sides matter here. A system with poor recall misses real danger, while a system with poor precision fills the dashboard with false alarms until the operator stops trusting it. The criteria in Table III have therefore been chosen to ask for a fair balance and not for a single high number.

TABLE III: PLANNED TESTS AND CRITERIA

Test	What it checks	Criterion
Count error	System count against hand count on test frames	Average error under 15%
Fire alert	Fire clips found; fire-like lights ignored	At least 90% found
Fall alert	Staged falls found; normal actions ignored	At least 85% found
Alert delay	Event in video to alert on dashboard	Under 5 s
SOS delivery	Pilgrim's SOS to admin screen	Under 3 s
Access control	Unapproved volunteer cannot log in	No bypass in all trials

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

The list of planned tests is short on purpose. A pilot must be easy to repeat, and every row of Table III can be repeated by another person with the same clips and the same counting sheets. If a criterion is not met, the result will be reported as it is and the cause will be studied, for instance a poor camera angle, a missing type of training picture or an overloaded server.

B. Comparison with Existing Work

Set against the reviewed work, the proposed system pulls several pieces together. The framework in [2] has monitoring and emergency reporting, but it relies on a cloud AI platform and lists no fire or fall module. The studies in [6] and [7] cover one danger each, and [3]–[5] are model studies with no user-facing side. The proposed design joins these parts on one platform that is meant to run on open-source tools and ordinary hardware. It does not match the face recognition and crowd-flow forecasting offered by [2], and it does not claim to replace the large control-room systems used at Prayagraj [1].

C. Limitations

Six limits should be stated plainly. First, counting accuracy drops in very dense scenes and with poor camera angles, rain or darkness. Second, live analysis of many streams needs a strong computer, which is why uploaded footage comes first. Third, fall detection works best when people are not packed tightly, because bodies hide one another, so it is planned for lanes, gates and medical areas and not for the densest bathing areas. Fourth, fire detection can be fooled by bright lights, a problem described in [6], so every alert must be confirmed by a person on the dashboard. Fifth, the website can slow down when very many pilgrims use it at once, so the pages are kept light. Sixth, video of people raises privacy questions. The system therefore stores counts and a few incident snapshots instead of full recordings, does not identify faces and keeps data only as long as the authorities allow. It is an assistant for the control room and not a replacement for it.

D. Expected Benefits for Each User Group

For the control room, the system turns a wall of video into a short ordered list of items that need attention, which reduces the strain on the operators. For volunteers, it replaces word-of-mouth instructions with a clear task and a clear status, and it lets the admin see who is free. For pilgrims, it offers a map of help points and a direct way to ask for help, which is useful for families who have been separated, for older visitors and for anyone who feels unwell. For the authorities, the stored records show where and when trouble occurred, which supports planning for the next Kumbh.

E. Responsible Use

A system of this kind must be used with care. The alerts are suggestions and a trained person must decide what to do with them. Crowd counts should be used to guide people and to open extra routes and not to blame or to single out any person. The authorities should decide in advance who may see the snapshots and for how long they are kept, and pilgrims should be told in simple Hindi and Marathi that cameras are used for safety. With these rules in place, the system can support the staff without becoming a source of new worry for the visitors.

F. Scalability and Maintenance

A system used during a festival has an unusual pattern of use. It may sit almost idle for months and then face very heavy traffic for a few weeks. For this reason the design keeps the AI unit and the web pages apart, so that either of them can be moved to a stronger machine when the event approaches. Zone names, limits and map points are stored as data and not written into the program, which lets the admin set up a new area without the help of a programmer. The models will need to be checked from time to time with new footage, especially after the seasons change, because rain, fog and different festival decorations can alter the way a scene looks to a detector. A short log of false alarms and missed events, kept by the admin, will show when a model needs to be retrained.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 3, November 2026
Impact Factor: 8.2

V. CONCLUSION

This work reviewed seven sources on crowd management, crowd counting, fire detection and fall detection, and used them to shape a web-based safety system for the Nashik Kumbh Mela. The review found that large deployments [1] and research frameworks [2] prove that AI can help, that counting models [3]–[5] and fire and fall studies [6], [7] work well in their own areas, and that none of them joins monitoring, volunteer work and pilgrim help in one place. The proposed system does this with Python, Django, OpenCV and deep-learning models, with Firebase or SQLite for data and Bootstrap for the screens.

Future work will build the full prototype and run the tests listed in Table III on real and staged video. After that, the next steps are to add the exact GPS position to every SOS, to offer Marathi and Hindi screens, to add crowd-growth forecasting and lost-and-found support and to carry out a small field trial with the help of the local authorities. If those trials show that the targets can be met, a system of this kind could give the control room, the volunteers and the pilgrims a common view of what is happening at the fair, and could help the authorities react earlier to danger.

Beyond the technical side, the main value of the proposal is practical. It does not ask the authorities to replace their cameras or their control rooms. It adds a thin layer on top of what already exists, so that the same footage reaches the right person sooner and a pilgrim in difficulty has a simple way to be heard. The design is open to change, and the lessons of the first trial are expected to shape the next version more than any assumption made on paper.

REFERENCES

[1] “AI in Maha Kumbh 2025,” Drishti IAS, Jan. 22, 2025. [Online]. Available: https://www.drishtiias.com/state-pcs-current-affairs/ai-in-maha-kumbh-2025

[2] R. R. Choudhury, A. Karmakar and R. P. Mitra, “Drishti AI-Event Guardian: An intelligent real-time crowd monitoring and emergency response system for mass gathering events,” arXiv:2606.05185, Jun. 2026.

[3] Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 589–597, doi: 10.1109/CVPR.2016.70.

[4] Y. Li, X. Zhang and D. Chen, “CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2018, pp. 1091–1100, doi: 10.1109/CVPR.2018.00120.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[6] A. Abdusalomov, N. Baratov, A. Kutlimuratov and T. K. Whangbo, “An improvement of the fire detection and classification method using YOLOv3 for surveillance systems,” Sensors, vol. 21, no. 19, Art. no. 6519, Sep. 2021, doi: 10.3390/s21196519.

[7] J. Gutiérrez, V. Rodríguez and S. Martin, “Comprehensive review of vision-based fall detection systems,” Sensors, vol. 21, no. 3, Art. no. 947, Jan. 2021, doi: 10.3390/s21030947.

[8] Django Software Foundation, “Django documentation.” [Online]. Available: https://docs.djangoproject.com/ (accessed Oct. 27, 2026).