SteelCount Pro: AI-Powered Steel Rod Detection and Inventory Management System Using Computer Vision and Progressive Web Application

Malladi Dhana Sai[1], Matcha Abhishek[2], Thota Charan[3], Achanta Siva[4], Malim Takhi Abbas[5]
[12345] Department of Computer Science and Engineering (AI & ML)
Aditya College of Engineering and Technology (Autonomous)
Surampalem, Kakinada District, Andhra Pradesh – 533437, India
Guide: Mr. R. S. V. V. Prasada Rao, M.Tech., Assistant Professor

Abstract

Steel rod inventory management at construction sites and warehouses relies on manual counting methods that are time-consuming (30–60 minutes per bundle) and error-prone (15–20% miscounts), leading to significant financial losses in the Indian construction industry. This paper presents SteelCount Pro, an AI-powered Progressive Web Application that automates steel rod detection, counting, size classification, and cost estimation from a single photograph using computer vision. The system employs a Roboflow-trained object detection model (Version 12) achieving 85–99% detection accuracy across eight IS 1786 standard rod sizes (6mm to 32mm). A novel inventory calculation pipeline automatically classifies detected rods by diameter, computes individual and total weights using the IS 1786 standard formula $W = D^2/162 \times L$, and generates instant cost estimates using configurable per-size pricing. The application supports two input modes (live camera capture and gallery upload), maintains a local scan history of up to 50 records with JSON export capability, and operates in an offline demo mode for environments with limited connectivity. Built with React 19 and TypeScript as a PWA with Capacitor for native Android/iOS deployment, the system reduces counting time by 90% and virtually eliminates human counting errors. Validated through 18 functional test cases, the platform provides a practical, zero-cost solution for construction inventory management.

Keywords: Steel Rod Detection, Object Detection, Computer Vision, Roboflow API, Progressive Web App, Inventory Management, Construction Technology, IS 1786

1. Introduction

Steel rods (rebars) are fundamental components in reinforced concrete construction, providing the tensile strength necessary for structures such as buildings, bridges, and infrastructure projects. The Indian construction industry, valued at over Rs. 12 lakh crore annually, consumes approximately 100 million tonnes of steel, with rebars constituting a substantial portion [1]. Accurate counting and inventory management of steel rods is essential for cost estimation, material procurement, quality control, wastage minimization, and theft prevention at construction sites and warehouses.

Despite their critical importance, steel rod inventory processes remain largely manual. Workers physically count rods in bundles containing 50–200 units, a task requiring 30–60 minutes per bundle with error rates of 15–20% [2]. The weighbridge alternative, which estimates counts from total bundle weight, requires expensive permanent installations (Rs. 5–15 lakh) and fails for mixed-size bundles—a common occurrence when deliveries contain multiple rod diameters. Spreadsheet-based tracking and basic tally counter apps provide no automation, no size classification, and no integrated cost calculation.

Recent advances in deep learning-based object detection have demonstrated the feasibility of automated counting and classification in industrial settings. YOLO (You Only Look Once) [3] and Faster R-CNN [4] architectures achieve real-time detection with accuracy exceeding 90% on trained object categories. Cloud-based platforms such as Roboflow [5] have democratized custom model training, enabling domain-specific detection models without deep ML expertise.

This paper presents SteelCount Pro, an AI-powered Progressive Web Application that addresses all limitations of existing methods. The key contributions of this work are:

- An end-to-end pipeline from image capture to inventory report using a Roboflow-trained detection model achieving 85–99% accuracy across 8 IS 1786 standard rod sizes.
- An automated inventory calculation engine computing per-size count, weight ($D^2/162 \times L$ formula), and cost with configurable pricing for all 8 standard sizes.
- A Progressive Web Application architecture with offline demo mode, local scan history, JSON export, and native Android/iOS deployment via Capacitor.
- Empirical validation demonstrating 90% reduction in counting time with near-zero counting errors compared to manual methods.

2. Related Work

Table 1 presents a comparative analysis of existing approaches to steel rod and construction material inventory management.

Table 1: Comparative Analysis of Related Work

S.No	Title / System	Year	Approach	Limitations	Ref
1	YOLO for Construction Objects (IEEE ICIP)	2021	Real-time object detection using YOLO for construction materials	Requires large training data, struggles with overlapping objects	[3]
2	Faster R-CNN (Ren et al.)	2017	Region proposal network for accurate object detection	Slower than YOLO, GPU-dependent for real-time	[4]
3	Roboflow Custom Detection	2022	Cloud-based model training and API deployment	API dependency, requires internet for live detection	[5]
4	Mobile CV for Inventory (ACM MobiSys)	2020	Smartphone cameras for inventory detection tasks	Lighting sensitivity, camera quality varies across devices	[6]
5	CNN Rebar Detection (Construction Eng.)	2021	CNN-based detection achieving 90%+ accuracy for rebars	Requires consistent image quality, no cost integration	[7]
6	TensorFlow Lite Mobile ML (Google AI)	2023	On-device ML inference for offline AI capability	Model size constraints, limited to simpler architectures	[8]

7	PWA for Industrial Apps (IEEE Software)	2022	Progressive Web Apps with native-like experience	Limited hardware access compared to native apps	[9]
8	SteelCount Pro (This Work)	2026	Roboflow detection + IS 1786 weight/cost engine + PWA + offline demo	API-dependent for live detection, no offline AI	-

The literature reveals three key gaps. First, existing rebar detection systems [7] focus exclusively on counting without integrating size classification, weight computation, or cost estimation into a unified pipeline. Second, cloud-based detection platforms [5] provide AI capabilities but lack domain-specific inventory features such as IS 1786 standard compliance, configurable pricing, and scan history. Third, no existing system combines AI-powered rod detection with a Progressive Web Application architecture that enables cross-platform deployment, offline functionality, and zero-infrastructure operation.

3. System Architecture and Design

SteelCount Pro follows a client-server architecture where the React 19 PWA handles user interaction, image capture, inventory calculation, and local data storage, while the Roboflow Cloud provides AI-powered object detection. Fig. 1 illustrates the system architecture.

Fig. 1: System Architecture (Client-Server with Roboflow Cloud)

The system workflow follows five sequential stages: (1) Capture—the user photographs the rod bundle using the device camera or uploads an existing image; (2) Analyze—the image is encoded as base64 and sent to the Roboflow detection API; (3) Classify—detected rods are normalized to IS 1786 standard sizes; (4) Calculate—per-size count, weight, and cost are computed; (5) Save—results are stored in local scan history with the original image.

3.1 Detection Pipeline

The detection pipeline processes images through the Roboflow API endpoint for the custom-trained model (inventory-of-steel-rods-um0a3, Version 12). The API receives a base64-encoded image via HTTPS POST request with two configurable parameters: confidence threshold (default 0.75, minimum detection certainty) and overlap threshold (default 0.50, maximum bounding box overlap to prevent double-counting). The API returns a JSON predictions array where each prediction contains: class (rod size label), confidence (float 0–1), and bounding box coordinates (x, y, width, height).

Algorithm 1 presents the complete detection-to-inventory pipeline.

Algorithm 1: Detection-to-Inventory Pipeline

```
Input: Image I, APIKey K, Prices P
Output: Inventory[], Summary

1. base64 ← encode(I)
2. IF size(base64) > 10MB THEN
     base64 ← resize(base64)
3. IF K = null THEN
     detections ← generateDemo()
   ELSE
     response ← POST(roboflow_url,
       base64, K, conf=0.75,
```

```
    overlap=0.50)
  detections ← response.predictions
4. FOR EACH size IN unique(det.class)
  items ← filter(det, class=size)
  count ← length(items)
  weight ← (D² / 162) × 12
  cost ← count × P[size].price
  totalWt ← count × weight
  inventory.add(size, count,
    weight, cost, totalWt)
5. summary ← sum(inventory)
6. RETURN (inventory, summary)
```

3.2 Weight Calculation (IS 1786 Standard)

The weight per rod is computed using the Bureau of Indian Standards formula from IS 1786:2008 [10]: $W = (D^2 / 162) \times L$, where D is the rod diameter in millimeters and L is the standard rod length of 12 meters. Table 2 presents the specifications for all 8 supported sizes.

Table 2: IS 1786 Rod Specifications and Default Pricing

Size	Price (Rs.)	W/m (kg)	W/12m Rod (kg)	Formula: D²/162
6mm	25	0.222	2.664	36/162 = 0.222
8mm	45	0.395	4.740	64/162 = 0.395
10mm	65	0.617	7.404	100/162 = 0.617
12mm	95	0.888	10.656	144/162 = 0.888
16mm	165	1.580	18.960	256/162 = 1.580
20mm	260	2.470	29.640	400/162 = 2.470
25mm	410	3.850	46.200	625/162 = 3.858
32mm	670	6.310	75.720	1024/162 = 6.321

4. Implementation

Table 3 presents the technology stack used in SteelCount Pro.

Table 3: Technology Stack

Layer	Technology	Purpose
Frontend	React 19 + TypeScript	Component-based UI with type safety
Build Tool	Vite 5.x	Fast bundling + Hot Module Replacement
Styling	Tailwind CSS + Radix UI	Responsive design + accessible primitives
AI Detection	Roboflow API (v12 model)	Custom-trained steel rod detection

Native Bridge	Capacitor	Android APK + iOS IPA builds
Storage	Browser localStorage	API key, prices, scan history (50 max)
PWA	Service Worker + Manifest	Offline caching, home screen install
Hosting	Netlify / Vercel / GitHub Pages	Free static hosting, zero backend cost

The application is organized into five modules: Camera Module (device camera initialization, permission handling, flash toggle, front/back switching, gallery upload), Detection Module (Roboflow API communication, response parsing, classification normalization, demo mode fallback), Inventory Module (grouping by size, weight calculation, cost computation, summary generation), Storage Module (localStorage CRUD for API key, prices, and scan history with 50-entry cap), and UI Module (five views: Home, Camera, Results, History, Settings with dark/light theme support).

All user data is stored locally using the browser's localStorage API. Three keys are used: roboflow_api_key (encrypted string), rod_prices (JSON object with price and weight per size), and scan_history (JSON array of scan records, each containing a UUID, ISO-8601 timestamp, base64 image thumbnail, detections array, inventory breakdown, and summary totals). No backend server is required, ensuring complete data privacy and zero operational cost.

4.1 Source Code: Inventory Calculation

```
// utils/inventory.ts
function calculateInventory(
  detections: Detection[],
  prices: PriceConfig
): InventoryResult {
  const grouped = groupBySize(detections);
  const items = Object.entries(grouped)
    .map(([size, rods]) => ({
      size,
      count: rods.length,
      weightPerRod: prices[size].weight,
      pricePerRod: prices[size].price,
      totalWeight: rods.length
        * prices[size].weight,
      totalCost: rods.length
        * prices[size].price,
      avgConfidence: average(
        rods.map(r => r.confidence))
    }));
  return {
    items,
    summary: {
      totalRods: sum(items, 'count'),
      totalWeight: sum(items,'totalWeight'),
      totalCost: sum(items, 'totalCost')
    }
  };
}
```

5. Results and Discussion

5.1 Detection Performance

Table 4 presents the performance metrics achieved by SteelCount Pro across key operational requirements.

Table 4: Performance Metrics

Metric	Target	Achieved	Status
Detection Accuracy	> 85%	85–99%	✓ Met
Detection Time (per image)	< 10 sec	3–5 sec	✓ Met
Image Preprocessing	< 2 sec	< 1 sec	✓ Met
Inventory Calculation	< 500ms	< 100ms	✓ Met
Page Load Time	< 3 sec	~2 sec	✓ Met
Counting Time Reduction	> 80%	90%	✓ Met
Human Error Elimination	> 90%	~100%	✓ Met

5.2 Sample Detection Output

To demonstrate the system's effectiveness, we present a sample detection from a real construction site image containing mixed rod sizes:

Table 5: Sample Detection Output (153 Rods Detected)

Size	Qty	Wt/Rod (kg)	Total Wt (kg)	Rate/Rod (Rs.)	Subtotal (Rs.)
16mm	54	18.96	1,023.84	1,232	66,550
20mm	93	29.63	2,755.59	1,926	1,79,113
25mm	6	46.30	277.80	3,010	18,057
TOTAL	**153**	-	**4,057.23**	-	**2,63,720**

The system processed this 2.12 MB image in 3.2 seconds, correctly identifying 153 rods across three sizes with an average confidence of 97.8%. The automatic size classification correctly distinguished 16mm rods from 20mm and 25mm rods based on bounding box radius mapping to IS 1786 standards. The total cost of Rs. 2,63,720 was computed instantly using the configured rate of Rs. 65/kg with 12m rod length.

5.3 Comparison with Existing Methods

Table 6: Comparison with Existing Inventory Methods

Feature	Manual	Weighbridge	Spreadsheet	Tally App	Ours
Counting Time	30–60 min	5–10 min	30+ min	20+ min	< 5 sec
Error Rate	15–20%	10–30%	5–10%	10–15%	1–5%
Size Classification	No	No	Manual	No	Auto

Weight Calculation	Manual	Measured	Formula	No	Auto
Cost Estimation	Manual	No	Manual	No	Auto
Photo Evidence	No	No	No	No	Yes
Mixed Size Support	Yes	Fails	Manual	Yes	Auto
Equipment Cost	Rs. 0	5–15 lakh	Rs. 0	Rs. 0	Rs. 0
Export / Audit Trail	No	No	Partial	No	JSON

5.4 Testing

Eighteen functional test cases validate the complete system workflow. Tests cover application loading, camera permission and initialization, image capture and file upload, API detection with valid key, demo mode without key, price display accuracy, weight calculation correctness (verified against IS 1786 formula for each of the 8 sizes), cost calculation accuracy, scan history CRUD operations (save, view, delete), JSON export, price configuration editing, offline mode, PWA installation, and theme toggling. All 18 test cases passed successfully.

6. Conclusion

This paper presents SteelCount Pro, an AI-powered Progressive Web Application that automates steel rod inventory management through computer vision-based detection, IS 1786-compliant size classification, and instant weight and cost estimation. The Roboflow-trained detection model achieves 85–99% accuracy across 8 standard rod sizes, processing images in 3–5 seconds. The system reduces manual counting time by 90% (from 30–60 minutes to under 5 seconds) and virtually eliminates the 15–20% human error rate.

The PWA architecture with Capacitor enables cross-platform deployment on web, Android, and iOS without maintaining separate codebases. The offline demo mode ensures functionality in low-connectivity construction environments. All data is stored locally with zero backend dependency, resulting in zero operational cost. The platform is validated through 18 functional test cases covering the complete detection, calculation, and storage workflow.

Future work includes TensorFlow.js integration for on-device offline AI detection, extending the model to detect rust and surface damage, PDF/Excel report generation, GPS location tagging for site-specific inventory tracking, barcode/QR integration for material tracking, and Firebase-based multi-user cloud synchronization for enterprise deployment.

References

[1] Indian Brand Equity Foundation, "Indian Construction Industry Overview," IBEF Report, 2024.

[2] S. Patel and R. Kumar, "Analysis of Manual Counting Errors in Construction Material Inventory," Indian Journal of Construction Management, vol. 12, no. 3, pp. 45–52, 2022.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proc. IEEE CVPR, pp. 779–788, 2016.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE TPAMI, vol. 39, no. 6, pp. 1137–1149, 2017.

[5] Roboflow Documentation, "Custom Object Detection Training and Deployment," Available: https://docs.roboflow.com, 2025.

[6] K. Chen, J. Li, and W. Deng, "Mobile Computer Vision for Industrial Inventory Management," Proc. ACM MobiSys, pp. 312–324, 2020.

[7] M. Zhang, L. Wang, and Y. Liu, "CNN-Based Steel Rebar Detection and Counting in Construction Sites," Construction Engineering and Management Journal, vol. 147, no. 8, 2021.

[8] Google AI, "TensorFlow Lite for Mobile and Edge Devices," Available: https://www.tensorflow.org/lite, 2023.

[9] A. Biorn-Hansen, T. Majchrzak, and T. Gronli, "Progressive Web Apps for Industrial Applications," IEEE Software, vol. 39, no. 5, pp. 68–74, 2022.

[10] Bureau of Indian Standards, "IS 1786:2008 High Strength Deformed Steel Bars and Wires for Concrete Reinforcement," BIS, New Delhi, 2008.

[11] R. Singh, P. Sharma, and V. Gupta, "YOLO-Based Construction Material Detection and Classification," Proc. IEEE ICIP, pp. 1234–1238, 2021.