Hand Gesture Recognition Model Performance Evaluation Report

Akash Sarode

Abstract

This report presents an evaluation of a custom hand gesture recognition model implemented using TensorFlow.js. The model was trained to classify 10 different hand gestures and achieved an overall accuracy of 93.7% on the test dataset. This document includes detailed performance metrics, confusion matrix analysis, confidence distributions, and ROC curve analysis across all gesture classes.

1 Model Overview

1.1 Model Development Process

The development of this custom hand gesture recognition model was a multistep process that involved data preprocessing, data collection, and deep learning techniques.

1.1.1 Data Normalization Algorithm

The foundation of the model begins with preprocessing hand pose data from TensorFlow's prebuilt "hand-pose-detection" model. A custom normalization algorithm was developed to standardize the raw 21 keypoint hand coordinates, ensuring consistent input regardless of hand size, position, or orientation in the camera frame.

The normalization process involves several linear transformations:

1. **Translation**: All keypoints are translated relative to the wrist position (keypoint 0), creating a coordinate system centered around the user's wrist.
2. **Rotation**: The hand is rotated using the palm direction vector (from wrist to middle finger MCP joint) to ensure consistent orientation.
3. **Scaling**: Hand size is normalized using the average distance from wrist to all metacarpophalangeal (MCP) joints.
4. **Vectorization**: The normalized coordinates are flattened into a 42 element feature vector (21 keypoints $\times$ 2 coordinates).

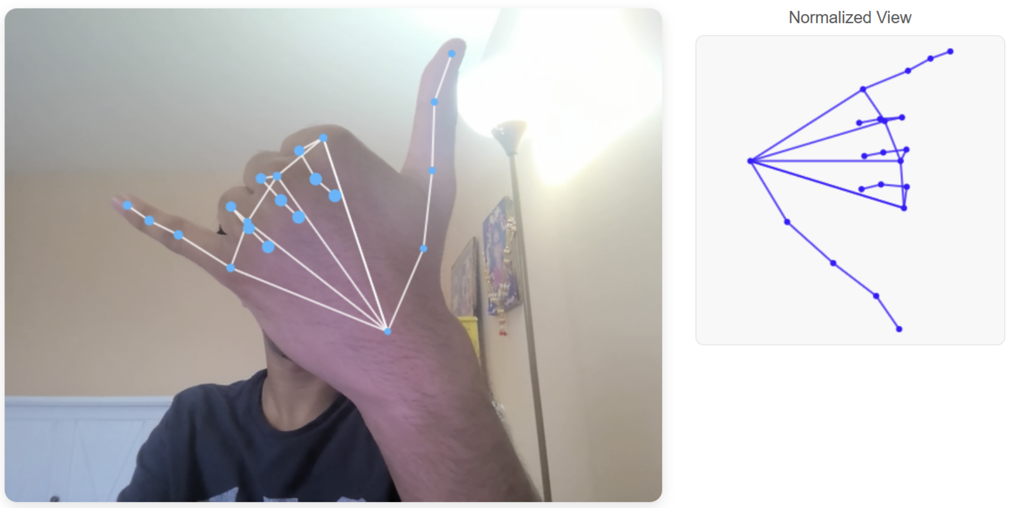
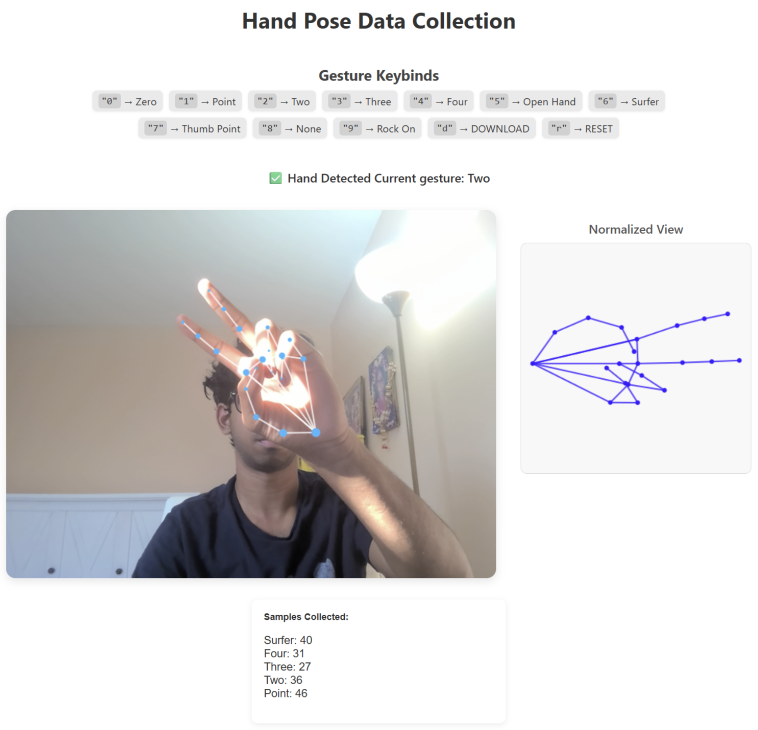


Figure 1: Visual representation of the data normalization algorithm transforming keypoint data.

This normalization ensures that gestures are recognized consistently regardless of the user's hand size, distance from camera, or orientation within the frame.

1.1.2 Data Collection

A custom Google Chrome extension was developed specifically for gesture data collection. This extension captured approximately 67,000 snapshots of normalized hand pose data across the 10 target gesture classes. The data was stored in a JSON format, with each sample containing a 42 element normalized feature vector and a corresponding gesture label.


Hand Pose Data Collection

Gesture Keybinds

"0" → Zero "1" → Point "2" → Two "3" → Three "4" → Four "5" → Open Hand "6" → Surfer
"7" → Thumb Point "8" → None "9" → Rock On "d" → DOWNLOAD "r" → RESET

✓ Hand Detected Current gesture: Two

Samples Collected:

Surfer: 40
Four: 31
Three: 27
Two: 36
Point: 46

Figure 2: Screenshot from the custom data collection Chrome extension. There are keybinds to record normalized data and label it as a specific gesture.

1.1.3 Model Architecture and Training

A custom deep neural network was implemented using TensorFlow.js for Node.js. The model architecture consists of:

- **Input Layer**: 42 features (normalized hand keypoint coordinates)
- **Hidden Layer 1**: 64 neurons with ReLU activation and 20% dropout
- **Hidden Layer 2**: 64 neurons with ReLU activation and 20% dropout
- **Output Layer**: 10 neurons with softmax activation (one per gesture class)

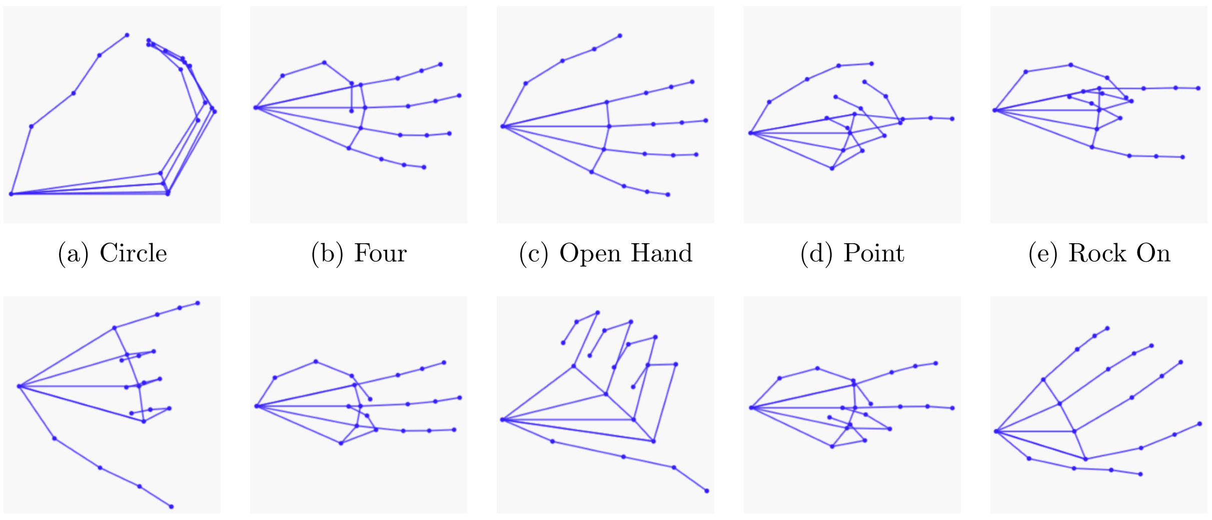
The model was trained using the Adam optimizer with categorical crossentropy loss over 30 epochs, using a batch size of 32 and 20% validation split. Dropout layers were utilized to prevent overfitting.

1.1.4 Model Evaluation

Following model training, a separate evaluation dataset was created containing approximately 20,000 samples (~2,000 per gesture class) to assess model performance. This evaluation set was used to generate the comprehensive performance metrics presented in this report.

1.2 Gesture Classes

The hand gesture recognition model was designed to classify the following 10 gesture classes. Each gesture demonstrates distinct finger positioning and hand orientation patterns that the normalized feature extraction algorithm can effectively distinguish.



(a) Circle (b) Four (c) Open Hand (d) Point (e) Rock On

(f) Surfer (g) Three (h) Thumb Point (i) Two (j) No Gesture

Figure 3: Complete set of 10 gesture classes recognized by the model accompanied with normalized keypoint images. The gestures include: circular finger formation (Circle), extended fingers (Four, Three, Two), directional pointing (Point, Thumb Point), symbolic gestures (Rock On, Surfer), full hand display (Open Hand), and various non-gesture positions (No Gesture). The "No Gesture" class represents scenarios where users are not actively performing a gesture, including instances where the hand is holding objects or in casual resting positions such as touching the user's face. The No Gesture image above was taken when a user was holding their phone.

2 Performance Metrics

2.1 Classification Report

Table 1 presents the detailed classification metrics for each gesture class.

Table 1: Classification Report - Performance Metrics by Class

Gesture Class	Precision	Recall	F1-Score	Samples
Circle	0.945	1.000	0.972	2170
Four	0.957	0.982	0.969	2011
No Gesture	0.800	0.893	0.844	2008
Open Hand	0.987	0.943	0.965	2011
Point	0.995	1.000	0.998	2011
Rock On	1.000	0.976	0.988	2014
Surfer	1.000	0.729	0.843	2011
Three	0.944	0.923	0.933	2006
Thumb Point	0.804	0.955	0.873	2023
Two	1.000	0.964	0.981	2002
Accuracy			**0.937**	**20267**
Macro Avg	**0.943**	**0.936**	**0.937**	**20267**
Weighted Avg	**0.943**	**0.937**	**0.937**	**20267**

2.2 Key Performance Insights

- **Overall Accuracy**: 93.7% - indicating strong model performance
- **Best Performing Classes**: Point (F1: 0.998), Rock On (F1: 0.988), Two (F1: 0.981)
- **Challenging Classes**: No Gesture (F1: 0.844), Surfer (F1: 0.843)
- **Perfect Precision**: Rock On, Surfer, and Two classes achieved 100% precision
- **Perfect Recall**: Circle and Point classes achieved 100% recall

3 Error Analysis

3.1 Most Common Misclassifications

Table 2 shows the top 5 most frequent misclassification patterns in the model.

Table 2: Top 5 Misclassification Patterns

True Class → Predicted Class	Count	Error Type
Surfer → Thumb Point	464	Similar hand positions
Three → No Gesture	137	Gesture not detected
No Gesture → Circle	126	False positive detection
Thumb Point → No Gesture	91	Gesture not detected
Surfer → No Gesture	81	Gesture not detected

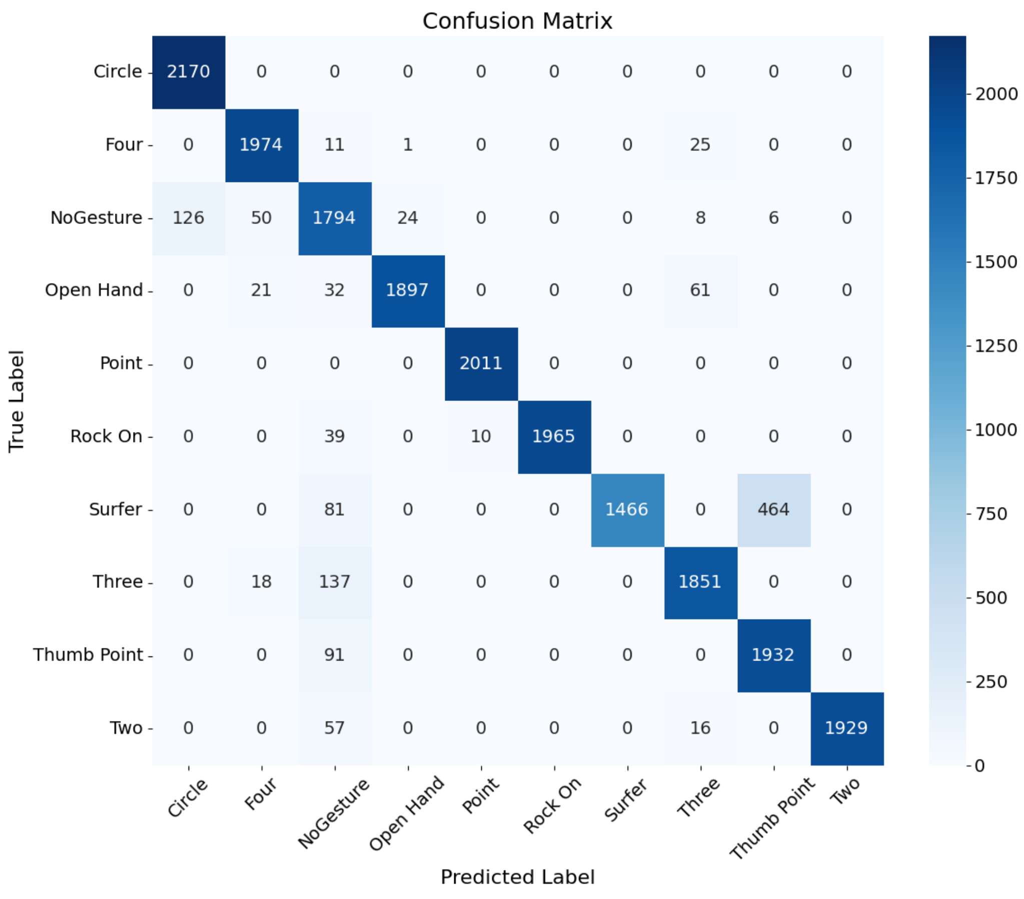
3.2 Error Pattern Analysis

The misclassification analysis reveals several key patterns:

1. **Surfer-Thumb Point Confusion**: The most common error (464 cases) involves confusing the "Surfer" gesture with "Thumb Point", likely due to similar finger positioning.
2. **Gesture Detection Issues**: Multiple classes (Three, Thumb Point, Surfer) are frequently misclassified as "No Gesture", indicating challenges in detecting subtle hand positions.
3. **False Positive Detection**: "No Gesture" samples are sometimes incorrectly classified as "Circle", suggesting the model may be oversensitive to circular hand motions.

4 Visual Analysis

4.1 Confusion Matrix



	Circle	Four	NoGesture	Open Hand	Point	Rock On	Surfer	Three	Thumb Point	Two
Circle	2170	0	0	0	0	0	0	0	0	0
Four	0	1974	11	1	0	0	0	25	0	0
NoGesture	126	50	1794	24	0	0	0	8	6	0
Open Hand	0	21	32	1897	0	0	0	61	0	0
Point	0	0	0	0	2011	0	0	0	0	0
Rock On	0	0	39	0	10	1965	0	0	0	0
Surfer	0	0	81	0	0	0	1466	0	464	0
Three	0	18	137	0	0	0	0	1851	0	0
Thumb Point	0	0	91	0	0	0	0	0	1932	0
Two	0	0	57	0	0	0	0	16	0	1929

Figure 4: Confusion Matrix showing the classification performance across all 10 gesture classes. The diagonal elements represent correct classifications, while off-diagonal elements indicate misclassifications. Darker blue colors indicate higher values.

4.2 Model Confidence Analysis

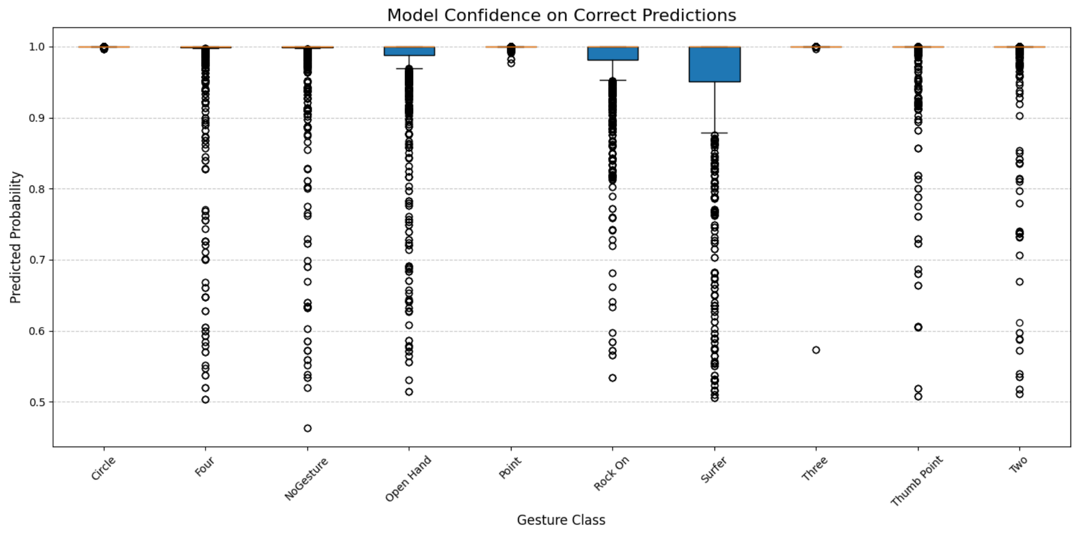


Figure 5: Box plot showing the distribution of model confidence scores for correct predictions across all gesture classes. Higher confidence values indicate greater model certainty. The Surfer class shows the widest confidence distribution, correlating with its lower recall performance.

4.3 ROC Curve Analysis

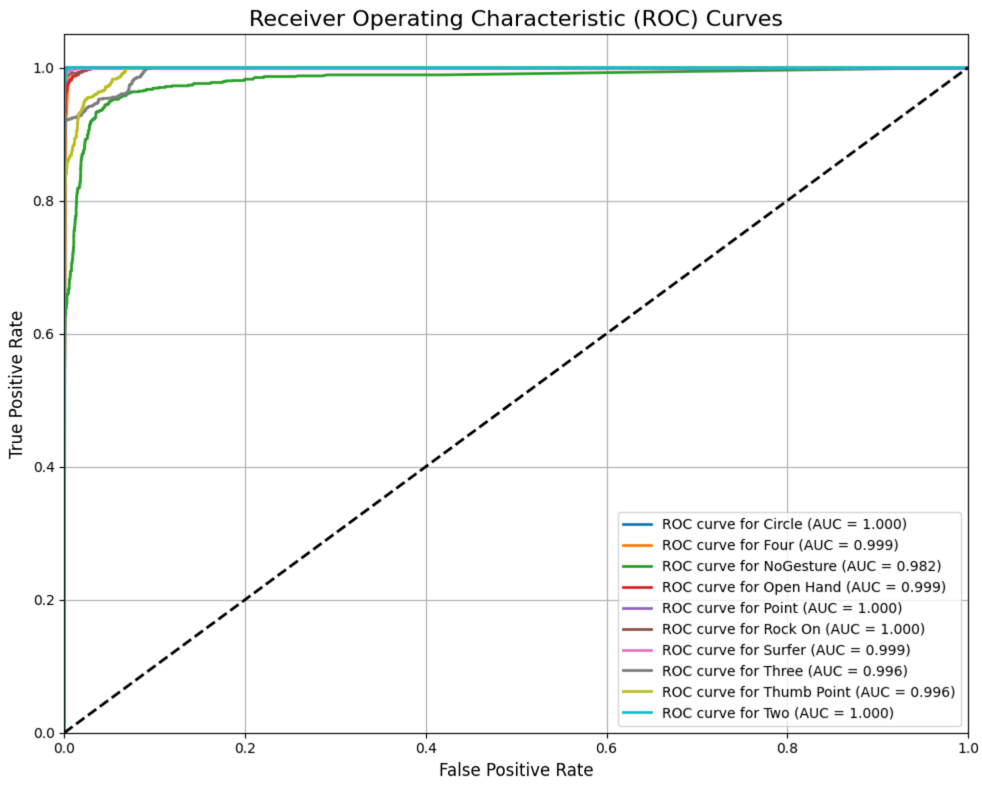


Figure 6: Receiver Operating Characteristic (ROC) curves for all 10 gesture classes. All classes demonstrate excellent discrimination ability with AUC values above 0.98. The curves clustered near the top-left corner indicate strong classifier performance with high true positive rates and low false positive rates.

5 Model Strengths and Limitations

5.1 Strengths

- High overall accuracy (93.7%) across 10 distinct gesture classes
- Excellent AUC scores (>0.98) for all classes indicating strong discriminative ability
- Perfect precision for Rock On, Surfer, and Two gestures
- Perfect recall for Circle and Point gestures
- Balanced performance across most gesture types

5.2 Areas for Improvement

- Surfer gesture classification shows lower recall (72.9%), frequently confused with Thumb Point
- No Gesture class has the lowest precision (80.0%), indicating false positive issues
- Need for better feature extraction to distinguish between similar hand positions
- Potential data augmentation for challenging gesture classes

6 Recommendations

1. **Data Augmentation**: Increase training samples for the Surfer class with varied hand orientations and positions
2. **Feature Engineering**: Implement additional hand landmark features to better distinguish between Surfer and Thumb Point gestures
3. **Threshold Optimization**: Adjust classification thresholds for the No Gesture class to reduce false positives
4. **Model Architecture**: Consider ensemble methods or attention mechanisms to improve performance on challenging gesture pairs
5. **Real-time Validation**: Test model performance in real-world scenarios with varying lighting and background conditions

7 Conclusion

The custom TensorFlow.js hand gesture recognition model demonstrates strong overall performance with 93.7% accuracy across 10 gesture classes. While most gestures are classified with high precision and recall, specific attention should be given to improving the distinction between similar gestures (particularly Surfer and Thumb Point) and reducing false positive detections for the No Gesture class. The high AUC scores across all classes indicate that the model has learned meaningful features for gesture discrimination, providing a solid foundation for deployment in hand gesture recognition applications.