SIGN LANGUAGE TRANSLATION SYSTEM USING COMPUTER VISION AND MACHINE LEARNING

Ayushi Shrote[1], Suyog Shende[2] , Prof. Ashwini Wakodikar[3]
[1,2]PG Scholar, [3]Assistant Professor Department of Computer Application
K.D.K.College of Engineering, Nagpur,Maharashtra, India
shroteasanjay.mca24f@kdkce.edu.in, suyoggshende.mca24f@kdkce.edu.in, ashwini.wakodikar@kdkce.edu.in

Abstract
Communication plays a vital role in human interaction. However, for hearing and speech-impaired individuals, sign language is the primary medium of communication. Unfortunately, the majority of people do not understand sign language, creating a communication gap between sign language users and non-users. To address this issue, this paper presents a Sign Language Translation System using Computer Vision and Machine Learning that translates hand gestures into understandable text output. The proposed system utilizes computer vision techniques to capture real-time hand gestures through a camera and processes them using image processing and machine learning algorithms. Hand landmarks are extracted and analyzed to recognize different gestures accurately. The recognized gestures are then translated into corresponding text, enabling effective communication. This system aims to provide an efficient, low-cost, and real-time solution to bridge the communication gap between hearing-impaired individuals and the general public. The proposed model demonstrates promising accuracy and can be further extended for real- world applications such as education, healthcare, and public services.
***Keywords** : Sign Language Translation, Hand Gesture Recognition, Computer Vision, Machine Learning, Human–Computer Interaction.*

INTRODUCTION

Sign language is a visually driven mode of communication in which meaning is expressed through structured hand movements, facial cues, and coordinated body actions. It is widely used by hearing and speech-impaired individuals as their primary mode of communication. Despite its importance, sign language is not commonly understood by the general population, which often leads to communication barriers in everyday situations such as hospitals, educational institutions, and public offices.

With advancements in computer vision and machine learning, automated sign language recognition systems have gained significant attention. These systems aim to recognize hand gestures and translate them into text or speech, enabling seamless communication. Hand gesture recognition is a key component of sign language translation as it focuses on detecting hand movements and shapes accurately. Vision-based approaches are preferred because they do not require any wearable devices and are more comfortable for users.

This paper proposes a Sign Language Translation System using Computer Vision and Machine Learning that captures hand gestures in real time, processes them using image processing techniques, and translates them into readable text. The system utilizes hand detection and landmark extraction methods to identify important hand features such as finger positions and palm orientation. These extracted features are analyzed using machine learning algorithms to classify different gestures.

The proposed system is designed to be simple, efficient, and suitable for real-time applications. It aims to reduce the communication gap between hearing-impaired individuals and the general public by providing an accurate and cost-effective translation solution. Additionally, the system can be extended in the future to support more complex gestures and continuous sign language translation.

LITERATURE REVIEW

Sign Language Recognition Systems

Several researchers have explored sign language recognition using sensor-based and vision-based approaches. Early systems relied on data gloves equipped with sensors to capture hand movements. Although accurate, such systems were expensive and uncomfortable for users. Vision-based systems later emerged as a cost-effective alternative, using cameras to capture gestures.

Recent studies employ machine learning and deep learning techniques such as Convolutional Neural Networks (CNNs) to recognize static and dynamic gestures. While these approaches show promising accuracy, many systems suffer from high computational complexity and limited real-time performance.

Hand Gesture Recognition Using Computer Vision

Hand gesture recognition involves detecting hand regions, extracting meaningful features, and classifying gestures. Techniques such as skin color segmentation, contour detection, and hand landmark extraction have been widely used. Hand landmark-based approaches reduce dimensionality while preserving essential gesture characteristics, resulting in improved performance.

Research Gap

Despite advancements, many existing systems are limited by lighting conditions, background noise, lack of real-time processing, and complex hardware requirements. There is a need for a simple, real-time, and affordable system that provides reliable gesture recognition with minimal setup. The proposed system aims to address these limitations.

PROPOSED SYSTEM ARCHITECTURE AND DESIGN

System Overview

The Sign Language Translation System using Computer Vision and Machine Learning is designed as a real-time, vision-based assistive communication system that translates hand gestures into meaningful textual output. The system follows a modular and sequential processing architecture to ensure accuracy, scalability, and real-time performance.

The overall workflow begins with capturing live video input through a standard camera device. The captured video stream is divided into individual frames, which are then analyzed to detect the presence of a human hand. Once the hand is detected, th e system extracts critical hand landmark points that represent finger positions, palm orientation, and relative joint movements. These extracted features serve as the primary input for the gesture recognition process.

System Modules

Input Acquisition Module: This module is responsible for capturing real-time video input using a camera device. The camera continuously records hand movements and converts the video stream into individual frames. These frames are forwarded to subsequent modules for further processing.The quality of input frames plays a crucial role in recognition accuracy. Therefore, the system supports adjustable camera resolution and frame rate to ensure optimal performance under different environmental conditions.

Preprocessing Module: The preprocessing module enhances raw input frames to improve gesture detection and recognition accuracy. This module performs operations such as:

Noise reduction to eliminate unwanted visual disturbances

Frame resizing to standardize input dimensions

Brightness and contrast normalization to handle lighting variations

Preprocessing ensures that the extracted features are consistent and reliable, reducing the impact of background noise and illumination changes.

Hand detection and Landmark Extraction Module: This module detects the presence of a hand within each video frame and extracts key landmark points representing the structure of the hand. These landmarks include finger joints, fingertips, and palm reference points.

Hand landmark extraction significantly reduces computational complexity by focusing only on essential gesture-related features instead of processing the entire image. The spatial relationships between landmarks provide a compact yet powerful representation of hand gestures.

Feature Extraction Module: The feature extraction module converts raw hand landmark data into numerical feature vectors suitable for machine learning algorithms. Features such as relative distances between fingers, angles between joints, and hand orientation are computed. By using relative measurements instead of absolute coordinates, the system becomes more robust to variations in hand size, position, and distance from the camera.

Gesture Recognition Module: The gesture recognition module classifies extracted feature vectors into predefined sign language gestures using machine learning techniques. During the training phase, the model learns gesture patterns from labeled datasets. During real-time execution, the trained model predicts the most probable gesture class.

This module plays a critical role in determining system accuracy and reliability. Efficient classification ensures real-time performance with minimal latency.

Text Translation and Output Module: Once a gesture is recognized, the system maps it to its corresponding textual representation using a predefined gesture-to-text dictionary. The translated text is displayed on the graphical user interface in real time. This module ensures that recognized gestures are presented in a clear and understandable format, enabling effective communication between sign language users and non-signers.

Graphical User Interface Module: The GUI module provides an interactive interface for users to view real-time gesture recognition results. It displays the live camera feed along with translated text output. The interface is designed to be simple and intuitive, allowing users with minimal technical knowledge to operate the system comfortably.

System Architecture

The proposed Sign Language Translation System follows a structured and modular architecture that integrates real-time visual processing with machine learning–based recognition. The architecture is designed to efficiently convert hand gestures into meaningful textual representations while maintaining low latency and high recognition accuracy. At the architectural level, the system operates as a pipeline in which visual data flows through successive processing layers. Each layer performs a dedicated function, ensuring clear separation of responsibilities and improved system scalability.

The architecture begins with a vision acquisition layer that captures continuous video input using a camera device. This visual stream acts as the primary source of information and is forwarded to the processing layer without manual intervention, enabling real-time interaction.

The processing layer is responsible for preparing the captured frames for analysis. It standardizes the visual input by applying frame normalization and color space conversion, allowing the system to handle variations in lighting and background conditions. This processed data is then passed to the hand analysis layer. In the hand analysis layer, key hand regions and spatial landmarks are detected and tracked. These landmarks represent the structural configuration of the hand and serve as the foundation for gesture understanding. By focusing on landmark-level representation instead of raw pixel data, the system reduces computational overhead while preserving critical gesture information.

The decision layer incorporates a trained machine learning model that evaluates the extracted landmark features. This layer performs gesture classification by mapping input features to predefined sign categories learned during the training phase. Finally, the output layer translates the recognized gesture into human-readable text and displays it through the user interface. This layer enables effective communication by presenting the interpreted meaning of the performed sign in real time. Overall, the architecture follows a sequential data flow:

Visual Input → Preprocessing → Hand Landmark Analysis → Feature Interpretation → Gesture Classification
→ Text Output.

This layered architectural design enhances clarity, supports future extensibility, and allows the integration of additional functionalities such as speech synthesis or multi-language translation without major structural changes.

Technical Stack

The proposed Sign Language Translation System is developed using a combination of computer vision libraries, machine learning frameworks, and programming tools to ensure efficient real-time gesture recognition and translation.

Programming Language:

Python is used as the primary programming language due to its extensive support for machine learning, computer vision, rapid application development.

Computer Vision Library:

OpenCV is employed for real-time video capture, frame preprocessing, and image manipulation tasks such as resizing, normalization, and region extraction.

Hand Landmark Detection Framework:

MediaPipe is utilized to detect and track hand landmarks accurately. It provides high-precision key points of fingers and palm regions, enabling reliable gesture representation.

Machine Learning Algorithms:

Supervised learning models such as Support Vector Machine (SVM) and Random Forest classifiers are used to classify hand gestures based on extracted landmark features.

Data Processing and Numerical Computation:

NumPy and Pandas are used for feature vector creation, data handling, and statistical operations.

User Interface:

Tkinter is used to design a simple graphical user interface (GUI) that allows users to interact with the system and view translated outputs.

METHODOLOGY AND SYSTEM DEVELOPMENT

Development Methodology

Data Collection: The first step in the methodology involves collecting hand gesture data representing commonly used sign language symbols. Gesture samples are captured using a standard camera in the form of images and video sequences. To improve system robustness, data is collected from multiple users with different hand sizes and gesture execution styles.

The dataset includes variations in hand orientation, distance from the camera, and slight changes in finger positioning. This diversity ensures that the trained model can generalize well to real-world scenarios rather than performing accurately only under ideal conditions.

Data Preprocessing: Raw video frames captured during data collection may contain noise, background distractions, and lighting inconsistencies. The preprocessing stage enhances input quality before feature extraction. Key preprocessing operations include:

Frame resizing to maintain uniform input dimensions

Noise reduction to remove visual disturbances

Normalization of brightness and contrast

Background suppression to emphasize hand regions

This stage plays a crucial role in improving the reliability and stability of subsequent processing steps.

Hand Detection and Landmark Localization: After preprocessing, the system identifies the presence of a hand within each frame. Once detected, precise hand landmarks such as fingertips, finger joints, and palm reference points are localized. These landmarks form a skeletal representation of the hand structure.

Landmark-based detection significantly reduces computational overhead by focusing on essential gesture-related features. It also enables accurate representation of complex hand postures using a limited set of spatial points.

Feature Extraction: In this stage, the localized hand landmarks are transformed into numerical feature vectors suitable for machine learning classification. Extracted features include:

Relative distances between fingertips and palm center

Angles between finger joints

Hand orientation and finger curvature patterns

By using relative geometric relationships instead of absolute pixel coordinates, the system becomes invariant to hand size, camera distance, and minor positional variations.

Model Training and Gesture Classification: The extracted feature vectors are used to train machine learning models capable of distinguishing between different hand gestures. During the training phase, labeled gesture data is fed into the classifier, allowing it to learn gesture-specific patterns.

Once trained, the model performs real-time gesture classification by predicting the most probable gesture class for each input frame. To reduce false predictions, temporal smoothing techniques are applied, ensuring that only consistent gesture predictions are accepted as valid outputs.

Gesture-to-Text Mapping: Each recognized gesture is mapped to a corresponding textual representation using a predefined gesture dictionary. This mapping ensures accurate translation of hand gestures into meaningful words or characters.

The mapping mechanism allows easy expansion of the gesture vocabulary by adding new gesture-text pairs without modifying the core recognition algorithm.

Output Generation and User Interaction: The final stage involves presenting the translated text to the user through a graphical user interface. The interface displays both the live camera feed and the recognized textual output in real time. This immediate feedback enables smooth interaction and ensures that users can communicate effectively without noticeable delay.

Performance Optimization and Validation: To ensure real-time operation, the system is optimized for low latency and efficient resource usage. Performance evaluation is carried out by testing the system under different lighting conditions, backgrounds, and gesture speeds.
Accuracy, response time, and stability are analyzed to validate the effectiveness of the proposed methodology. Continuous refinement of preprocessing parameters and feature selection contributes to improved system performance.

Requirements Analysis

Requirement analysis defines the foundational needs of the proposed Sign Language Translation System and serves as a guideline for its design and development. This phase focuses on identifying system expectations, operational boundaries, and performance goals to ensure effective gesture translation. The system is required to process visual input captured through a camera and recognize hand gestures in real time. Accurate identification of hand movements and their correct mapping to textual meaning is a core functional requirement. The system must continuously analyze incoming frames and provide timely output to support natural interaction.
Another essential requirement is ease of use. The system should offer a clear and simple interaction mechanism so that users can perform gestures without complex setup or configuration. The translated text output must be easily readable and updated dynamically. From a performance perspective, the system should maintain stable operation under different environmental conditions, such as changes in lighting or background. Efficient processing and low response time are necessary to ensure smooth real-time translation.
The system must also be flexible and extensible. It should allow the inclusion of additional gesture classes without significant modifications to the existing structure. Compatibility with commonly available hardware and software platforms is required to support wider usability. These requirements collectively influence architectural decisions, model selection, and implementation strategies, ensuring that the developed system meets both functional expectations and practical constraints.

System Design Process

The design of the proposed Sign Language Translation System is carried out through a systematic and goal-oriented process that bridges conceptual planning and practical deployment. The primary objective of the design process is to achieve reliable gesture interpretation while maintaining real-time responsiveness and structural clarity. The process begins with defining system objectives and constraints, focusing on accurate hand gesture recognition, minimal processing delay, and ease of user interaction. Based on these objectives, appropriate technologies and frameworks are identified to support vision-based analysis and machine learning–driven decision making.
In the subsequent phase, the system workflow is outlined by establishing how visual data moves through different functional components. This includes defining interactions between video acquisition, visual analysis, feature formulation, and classification units. A layered design approach is adopted to clearly separate data handling, computation, and output generation responsibilities. The gesture representation strategy is then designed by determining how hand movements and postures can be effectively translated into numerical features. Instead of relying on raw image data, the system emphasizes landmark-based representation to improve efficiency and robustness.
Following this, the learning mechanism is designed by selecting suitable classification models and training strategies. The design ensures that the model can generalize well across variations in hand orientation and user differences. Finally, the output interaction design is planned to present recognized gestures in an understandable textual format. The design process concludes with considerations for extensibility, allowing additional functionalities such as audio output or expanded gesture vocabularies to be incorporated without major architectural changes.

EXPERIMENTAL EVALUATION AND RESULTS

Evaluation Methodology

The evaluation methodology was designed to measure the performance of the proposed system across multiple parameters, including gesture recognition accuracy, response time, and consistency of predictions. A controlled experimental approach was adopted to ensure fairness and repeatability of results.

The system was evaluated using a predefined set of static hand gestures commonly used in sign language. Each gesture was performed multiple times by different users to analyze variation handling. For every input gesture, the predicted output was compared with the expected gesture label.

The primary evaluation metrics used in this study include:

Recognition Accuracy: Percentage of correctly identified gestures

Response Time: Time taken to generate textual output after gesture input

Stability of Prediction: Consistency of recognition across consecutive frames

By analyzing these metrics, the system's reliability and suitability for real-time assistive communication were assessed.

Experimental Setup

The experimental setup consisted of a standard laptop equipped with a built-in camera. The system was implemented using Python programming language along with computer vision and machine learning libraries. No external sensors or specialized hardware were used.

Experiments were conducted in an indoor environment with moderate lighting to minimize external interference. The camera was positioned at a fixed distance from the user to capture clear hand movements. Multiple users participated in the evaluation to ensure diversity in hand shape, gesture execution speed, and orientation.

The gesture dataset was divided into training and testing phases. During training, labeled gesture samples were used to train the machine learning model. During testing, unseen gesture inputs were provided to evaluate the system's prediction capability. To simulate real-world usage, users were allowed to perform gestures naturally without strict constraints on speed or hand orientation. This ensured that the evaluation results reflect realistic operating conditions.

Results and Analysis

The experimental results demonstrate that the proposed system performs effectively in recognizing static hand gestures and translating them into text in real time. The system achieved high recognition accuracy for predefined gestures under controlled conditions. It was observed that gestures with distinct finger configurations were recognized more accurately compared to gestures with similar hand structures.The use of hand landmark-based feature extraction contributed significantly to reducing misclassification by capturing precise finger and joint positions.

The response time of the system remained within acceptable limits, allowing near real-time interaction. Minor delays were observed only during rapid gesture transitions, which can be addressed in future enhancements using temporal filtering techniques. Performance analysis also revealed that stable lighting conditions improved recognition accuracy, while excessive background complexity occasionally affected detection. However, preprocessing and feature normalization helped mitigate these effects to a considerable extent.

Overall, the results validate that the proposed system is capable of delivering reliable and efficient sign language translation for assistive communication. The analysis confirms the feasibility of using vision-based hand gesture recognition as a practical solution for bridging communication gaps between hearing-impaired individuals and non-signers.

Performance Metrics

To evaluate the effectiveness of the proposed Sign Language Translation System, several performance metrics were considered. These metrics provide a quantitative understanding of the system's accuracy, efficiency, and real-time responsiveness, which are critical for practical deployment in assistive communication.

Accuracy

Accuracy measures the proportion of correctly recognized gestures relative to the total gestures performed. It provides an overall indicator of the system's ability to correctly interpret hand movements.

Response Time (Latency)

Response time represents the delay between the execution of a gesture and the display of the corresponding textual output. Minimizing latency is essential to ensure smooth, real-time interaction for users.

Frame Processing Rate (FPS)

The frame processing rate indicates the number of video frames the system can process per second. A higher FPS contributes to a seamless and continuous gesture recognition experience.

Robustness

Robustness evaluates the system's stability under varying environmental conditions, such as changes in lighting, background
complexity, and hand orientation. This metric ensures consistent performance across different users and scenarios.

The combination of these metrics allows for a comprehensive assessment of the system's technical performance, highlighting both its accuracy in gesture recognition and its suitability for real-time, practical applications. Evaluating the system across these parameters demonstrates its potential for reliable and user-friendly assistive communication.

TECHNICAL IMPLEMENTATION DETAILS

Frame Acquisition and Processing Pipeline

The system begins its execution by initializing the camera interface to capture continuous live video input. The incoming video stream is divided into individual frames, which are processed sequentially to maintain real-time interaction. To avoid unnecessary latency, the system prioritizes the most recent frames and discards outdated ones.

Each captured frame is standardized by resizing it to a fixed resolution. This normalization ensures consistency across different camera devices and enhances the reliability of downstream processing. Basic preprocessing operations such as noise reduction and brightness normalization are applied to enhance visual clarity.

After preprocessing, the system identifies the region of interest containing the user's hand. By focusing only on the hand region rather than the entire frame, the system significantly reduces computational overhead. This pipeline design ensures efficient frame handling while preserving the essential visual information required for gesture recognition.

Gesture Representation and Machine Learning-Based Recognition

Once the hand region is identified, the system applies computer vision techniques to extract meaningful structural information from the hand. Key landmark points such as finger tips, joint locations, and palm reference points are detected to form a skeletal representation of the hand.

These landmarks are converted into numerical feature vectors using geometric relationships such as relative distances between fingers, joint angles, and hand orientation. Using relative features instead of absolute pixel values makes the system robust to variations in hand size, position, and distance from the camera.

The machine learning component of the system uses supervised learning techniques. During the training phase, labeled gesture samples are used to learn distinctive patterns associated with each gesture. During real-time execution, extracted features are classified to predict the most probable gesture. To improve reliability, the system validates predictions across multiple consecutive frames. This reduces misclassification caused by accidental movements or incomplete gestures and ensuring stable recognition output.

System Integration and Runtime Optimization

All functional components of the system are integrated into a unified runtime framework that enables seamless data flow between modules. Each module operates independently but communicates through well-defined interfaces, improving maintainability and scalability.

A lightweight graphical user interface is implemented to display both the live video feed and the translated textual output. This interface provides immediate feedback to users and enhances usability without interfering with recognition performance.

Runtime optimization techniques such as selective frame processing, reduced feature dimensionality, and efficient memory management are applied to ensure low latency. These optimizations enable the system to run efficiently on standard computing devices, eliminating the need for high-end hardware or external sensors.

FUTURE ENHANCEMENTS

Support Dynamic and Continuous Gestures

The current implementation primarily focuses on recognizing static hand gestures. However, real-world sign language communication often involves dynamic and continuous gestures combined with temporal motion patterns. Future enhancements can incorporate temporal modeling techniques to recognize gesture sequences rather than isolated frames. Motion-based features and sequence learning approaches can be used to capture gesture transitions more effectively. This will allow the system to interpret complete words and sentences instead of individual signs, making communication more natural and expressive. Incorporating continuous gesture recognition will significantly enhance system usability in conversational scenarios and educational applications.

Advanced Learning Models and Accuracy Improvement

While traditional machine learning techniques provide reliable results, future versions of the system can integrate advanced learning models to improve recognition accuracy and adaptability. Deep learning architectures can automatically learn complex gesture patterns from large datasets without manual feature engineering.

Adaptive learning mechanisms can also be introduced, allowing the system to personalize recognition based on individual user styles. Continuous learning from user interaction can improve robustness against variations in hand posture, gesture speed, and environmental conditions. Such enhancements will increase system reliability and make it suitable for deployment in diverse real- world environments.

Multi-Modal Output and Platform Expansion

Future enhancements may extend the system beyond text output by incorporating additional output modalities. Speech synthesis can be integrated to convert recognized gestures into audible output, enabling communication with visually impaired individuals as well. Furthermore, the system can be extended to support multiple spoken languages, making it accessible to a broader user base. Deployment on mobile platforms and integration with web-based applications can improve portability and reach. These extensions will transform the proposed system into a comprehensive assistive communication platform suitable for widespread adoption.

CONCLUSION

This paper presented a vision-based Sign Language Translation System that applies computer vision and machine learning techniques to facilitate effective communication between sign language users and non-sign language users. The proposed approach focuses on interpreting hand gestures captured through a camera and converting them into meaningful textual representations in real time.

A landmark-driven gesture representation strategy is employed to capture essential hand motion and posture information while avoiding the limitations of pixel-level analysis. This design choice enables efficient processing and contributes to stable recognition performance across different users and environmental conditions. The layered architecture and modular workflow further enhance system clarity, maintainability, and ease of integration.

The experimental observations indicate that the system is capable of delivering responsive gesture recognition with minimal processing delay, making it suitable for practical, real-world scenarios. Additionally, the reliance on commonly available hardware highlights the feasibility of deploying the system without specialized or expensive equipment, thereby increasing its accessibility and potential for widespread adoption.

Beyond its current scope, the proposed system provides a strong foundation for future research and development. The architecture can be extended to support a larger vocabulary of gestures, sentence-level interpretation, and multimodal output such as speech synthesis. Furthermore, integration with advanced learning models and multilingual support can enhance the system's robustness and usability across diverse user groups.

Overall, this work contributes to the advancement of assistive communication technologies by demonstrating how intelligent visual systems can be effectively leveraged to address real-world communication challenges. The proposed solution underscores the role of scalable, real-time, and user-centric design in promoting accessibility and fostering inclusive human–computer interaction in modern digital environments.

REFERENCES

1. T. M. Mitchell, *Machine Learning*, McGraw-Hill, New York, USA, 1997.
2. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
3. S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 37, no. 3,
4. pp. 311–324, 2007.
5. T. Starner, J. Weaver, and A. Pentland, "Real-time American Sign Language recognition using vision-based systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1371–1375, 1998.
6. C. Keskin et al., "Hand pose estimation and recognition using computer vision," *ECCV*, pp. 852–863, 2012.
7. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
8. S. Zhang et al., "Vision-based hand gesture recognition for human–computer interaction," *Journal of Visual Communication and Image Representation*, vol. 25, no. 8, pp. 1914–1924, 2014.
9. G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
10. K. Simonyan and A. Zisserman, "Very deep convolutional networks for image recognition," *ICLR*, 2015.
11. Chaudhary et al., "Vision-based sign language recognition system," *International Conference on Signal Processing*, 2011.