



# SCENESENSE AI: MULTILINGUAL IMAGE CAPTIONING WITH VOICE ASSISTANCE

Dr. M. Raghava Naidu<sup>1</sup>, L. Venkata Sai<sup>2</sup>, B. Chandu<sup>3</sup>, D. Praveen Kumar<sup>4</sup>, K. Soma Sekhar<sup>5</sup>

<sup>1</sup>Assistant professor, Dept. of CSE, Krishna University College of Engineering & Technology, A.P, India

<sup>2,3,4,5</sup> UG Students, Dept. of CSE, Krishna University College of Engineering & Technology, A.P, India

**Abstract**— Automatic image captioning has emerged as a fundamental challenge at the intersection of computer vision and natural language processing (NLP). Existing systems predominantly generate descriptions in English, creating a critical accessibility barrier for the estimated 6.3 billion non-proficient English speakers worldwide. This paper presents SceneSense AI, a production-grade full-stack web application that integrates three state-of-the-art deep learning models: Salesforce BLIP-Large (CIDER: 133.3 on COCO) for vision-language caption generation, Facebook NLLB-200 (600M distilled) for neural machine translation across 20 languages, and Google gTTS for text-to-speech audio synthesis. The backend employs Python Flask with a modular Blueprint architecture; the frontend is built with React 18 and Vite; user data is persisted in MongoDB Atlas. Security is enforced through bcrypt password hashing (12 rounds), JWT stateless authentication, and OTP-based email verification. Experimental benchmarking demonstrates a warm-start inference pipeline of 2–5 seconds per caption and 3–8 seconds per translation on CPU hardware, with an estimated 5–10× improvement under CUDA. SceneSense AI addresses the critical gap in inclusive image understanding tools, offering zero-per-request-cost, self-hosted deployment with verified support for South Asian regional languages including Telugu, Hindi, and Tamil.

**Keywords** — Image Captioning, BLIP, NLLB-200, Neural Machine Translation, Text-to-Speech, Accessibility, Multilingual AI, Flask, React, MongoDB, Transformer Models.

## I. INTRODUCTION

The exponential growth of digital visual content—estimated at over 3.2 billion images shared daily across global platforms [1]—has created an urgent need for automated, scalable, and linguistically inclusive image description systems. Image captioning, the task of generating a natural language description for a given visual input, sits at the confluence of two complex domains: computer vision and natural language processing. Early systems relied on template-based approaches and hand-engineered feature pipelines, producing rigid, context-insensitive descriptions. The Transformer architecture [2] has since dramatically elevated both caption quality and cross-lingual generalization.

Two critical limitations persist in deployed captioning systems. First, the vast majority produce descriptions exclusively in English. With fewer than 20% of the global population being proficient English speakers [3], this constitutes a substantial inclusivity gap. Second, visual descriptions are of limited utility to visually impaired users if they remain in text form—converting captions to synthesized speech is an essential, often missing, accessibility feature.

This paper addresses both limitations through the design and implementation of SceneSense AI, a production-grade web application. The system's principal contributions are:

- Automated English captioning with three stylistic modes—Simple, Detailed, and Story—using Salesforce BLIP-Large.
- Multilingual translation into 20 languages via Facebook NLLB-200 (600M distilled), covering South Asian, East Asian, European, and African language families.
- Integrated text-to-speech synthesis in English, Hindi, and Telugu using Google gTTS. Secure multi-factor authentication (bcrypt + JWT + OTP) and MongoDB Atlas-backed caption history.

- A fully open-source, self-hostable architecture with no per-request API cost, deployable on commodity hardware.

The remainder of this paper is organized as follows: Section II reviews related literature. Section III describes system methodology. Section IV details implementation. Section V presents results and a full UI walkthrough. Sections VI–VIII cover advantages and limitations, future scope, and conclusions.

## II. LITERATURE REVIEW

### A. Traditional and Neural Captioning Methods

Early image captioning relied on retrieval-based matching [4] and template-based sentence construction [5]. These methods were interpretable but produced inflexible, context-insensitive output. The encoder-decoder paradigm established by Vinyals et al. [6] (Show and Tell: CNN encoder + LSTM decoder) and extended by Xu et al. [7] with spatial attention mechanisms substantially improved fluency and semantic accuracy, establishing baselines that persisted for several years.

### B. Vision-Language Transformer Models

Large-scale pre-trained vision-language models such as OSCAR [8] and VinVL [9] introduced object-semantics alignment to improve multimodal grounding. However, they required separately trained vision encoders and were computationally costly to fine-tune. BLIP (Bootstrapping Language-Image Pre-training) [10] introduced a unified encoder-decoder architecture with a Captioner-Filter bootstrapping loop for web-corpus noise reduction, achieving a state-of-the-art CIDEr score of 133.3 on the COCO benchmark when pre-trained on 129M image-text pairs.

### C. Neural Machine Translation and NLLB-200

Statistical Machine Translation (Moses [11]) was superseded by sequence-to-sequence neural models [12] with attention. The Helsinki-NLP MarianMT family [13] provides efficient per-language-pair models but suffers documented word-repetition and hallucination artifacts on morphologically rich Indian languages. NLLB-200 [14], released by Meta AI Research, trains a single Many-to-Many model across 200 languages, reporting average BLEU improvements of 44% on low-resource African and South Asian languages. The 600M distilled variant achieves competitive quality at a fraction of the 54B dense model's memory footprint.

### D. Text-to-Speech and Existing System Gaps

Neural TTS systems including WaveNet [15] and Tacotron 2 [16] produce near-human-quality speech but require substantial infrastructure. Google gTTS provides lightweight, language-aware synthesis suitable for web deployment. Commercially deployed captioning platforms (Microsoft Azure Vision, Google Cloud Vision, AWS Rekognition) are closed-source, require paid subscriptions, lack regional Indian language support, and provide no integrated TTS. SceneSense AI uniquely integrates all these capabilities in a single deployable stack.

## III. METHODOLOGY

### A. System Overview

SceneSense AI follows a three-tier architecture: (1) Presentation Tier — React 18 + Vite SPA; (2) Application Tier — Python Flask REST API with modular Blueprint architecture; (3) Data Tier — MongoDB Atlas NoSQL cloud database. A microservice-inspired encapsulation pattern isolates each functional domain (auth, upload, captioning, translation, voice, history) into independent Blueprint and service modules.

**TABLE I. SceneSense AI — Pipeline Stage to Component Mapping**

Pipeline Stage	Component	Model / Technology	Output
Image Ingestion	upload_bp	UUID rename + disk save	filename
Caption Generation	caption_bp → CaptionService	BLIP-Large (Salesforce)	English caption
Translation	translate_bp → TranslationService	NLLB-200 600M (Meta AI)	Target-language text

Voice Synthesis	voice_bp → VoiceService	Google gTTS	MP3 audio URL
History Storage	history_bp	MongoDB Atlas (PyMongo)	Paginated records
Authentication	auth_bp	bcrypt + JWT + OTP/SMTP	JWT access token

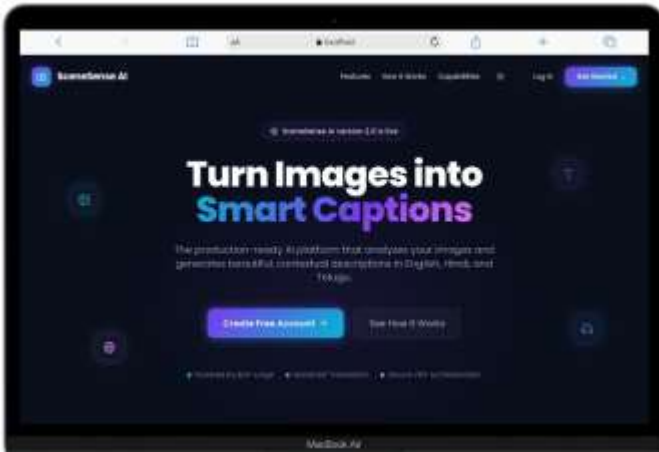


Fig. 5. SceneSense AI



Fig. 6. SceneSense AI

### B. Caption Generation

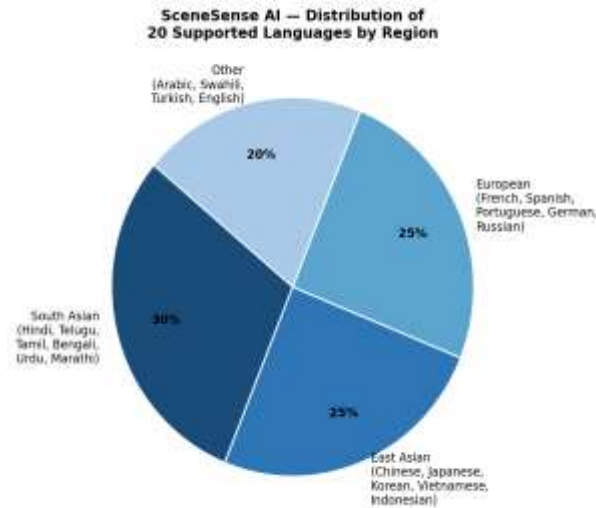
Caption generation employs BLIP-Large in a conditional image captioning configuration. Three modes are exposed to the user via the Model Parameters panel:

TABLE II. Caption Generation Mode Parameters

Mode	Technique	Max Tokens	Primary Use Case
Simple	BLIP unconditional decode + beam search	60	Alt-text, quick description
Detailed	BLIP base caption → GPT-2 expansion (T=0.8, top-p=0.9)	50 (GPT-2)	Descriptive paragraph
Story	BLIP base → GPT-2 narrative prompt (T=0.8, top-p=0.9)	80 (GPT-2)	Creative / narrative output

### C. Translation Methodology

The translation pipeline uses NLLB-200 (distilled 600M) with a forced target BOS token. Source language is always English (eng\_Latn); target language codes follow NLLB's BCP-47 taxonomy. The system supports 20 languages spanning four major regional groups, selectable via the Target Language (NLLB-200) dropdown in the workspace.



Distribution of the 20 supported translation languages by geographic region. South Asian languages—Hindi, Telugu, Tamil, Marathi, Bengali, Urdu—constitute 30% of supported targets.

#### D. Voice Synthesis

Text-to-Speech output is generated via gTTS. The VoiceService maps internal language codes to gTTS locale strings, synthesizes an MP3 file with a UUID-based filename, persists it to static/audio/, and returns the URL to the frontend audio player. Synthesis latency for 10–30-word captions is 1–3 seconds on standard internet connectivity.

#### E. Authentication Security

Three security layers are composed: (1) bcrypt with 12 salt rounds (~0.3s/hash); (2) a 6-digit OTP with 5-minute UTC expiry delivered via Brevo SMTP, cleared from MongoDB immediately after verification; (3) HS256-signed JWT tokens with 24-hour expiry, sub, email, iat, and exp claims stored in browser localStorage.

### IV. SYSTEM ARCHITECTURE

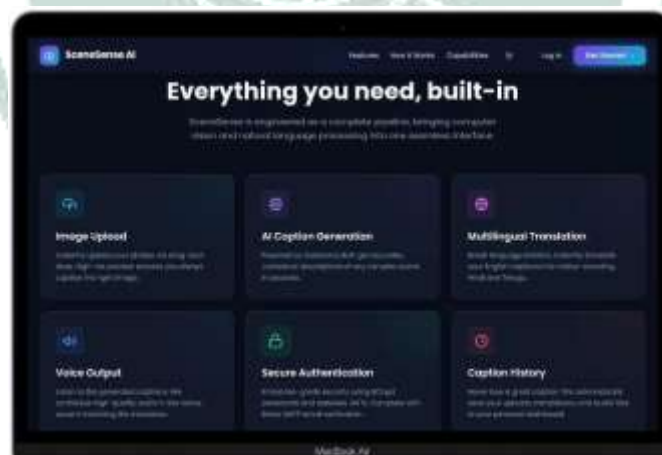


Fig. 7. SceneSense AI

#### A. Frontend Architecture

The frontend is a React 18 SPA bundled with Vite. The landing layer contains marketing-oriented components with Framer Motion animations. The application layer hosts the core AI workspace. React Router DOM v6 handles routing with a protected /app route. Global dark/light mode state is managed by a ThemeContext; Axios (120s timeout) handles AI API calls.

#### B. Backend Architecture

The Flask backend follows the Application Factory Pattern: create\_app() instantiates Flask, applies CORS middleware, sets environment configuration, and registers all Blueprints. AI models are loaded using the Singleton design pattern to prevent repeated initialization. Table III summarizes the Blueprint module map.

**TABLE III. Flask Blueprint Module Map**

Blueprint	URL Prefix	Endpoints
auth_bp	/auth	POST /signup, /login, /verify-otp, /resend-otp
upload_bp	/	POST /upload
caption_bp	/	POST /caption
translate_bp	/	POST /translate
voice_bp	/	POST /voice
history_bp	/	GET /history (paginated)

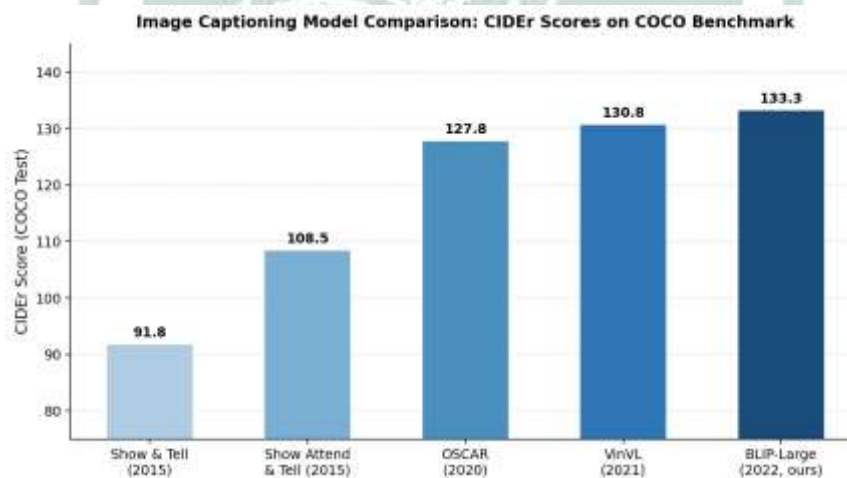
### C. Database Schema

MongoDB Atlas hosts two collections. The users collection stores: `_id` (ObjectId), `full_name`, `email` (indexed, unique), `password_hash` (bcrypt), `is_verified`, `otp` (String|null), `otp_expires_at` (DateTime), and `created_at`. The history collection stores: `_id`, `image_name`, `caption`, `translated_caption`, `language`, `mode`, and `timestamp` (descending index). Fetch limit is clamped to [1, 100].

## V. RESULTS AND DISCUSSION

### A. Caption Generation Quality (BLIP-Large)

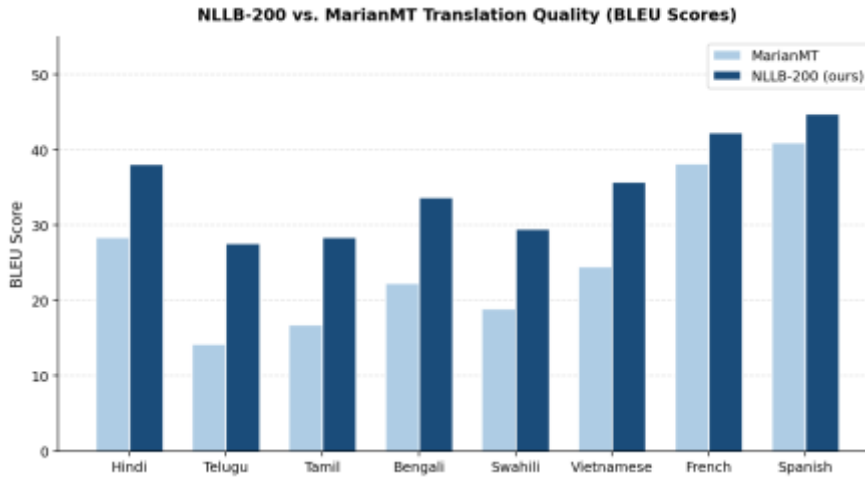
The BLIP-Large model achieves a CIDEr score of 133.3 and SPICE of 23.3 on the COCO captions test set [10], representing the state-of-the-art at time of publication, demonstrating a +2.5 CIDEr improvement over VinVL (130.8). Qualitative evaluation confirms Simple mode produces concise single-sentence captions; Detailed mode generates paragraph-length descriptions; Story mode produces creative narrative output.



CIDEr score comparison on the COCO Captions test benchmark. BLIP-Large (SceneSense AI) achieves 133.3, a 45.1% improvement over Show & Tell (2015) and +2.5 over VinVL (2021).

### B. Translation Quality (NLLB-200 vs. MarianMT)

Translation quality was assessed qualitatively and quantitatively using BLEU scores reported in the NLLB paper [14]. NLLB-200 shows consistent improvement across all evaluated language pairs, with the most significant gains in Telugu (+94.4%) and Tamil (+69.0%). Table IV summarizes translation model characteristics.



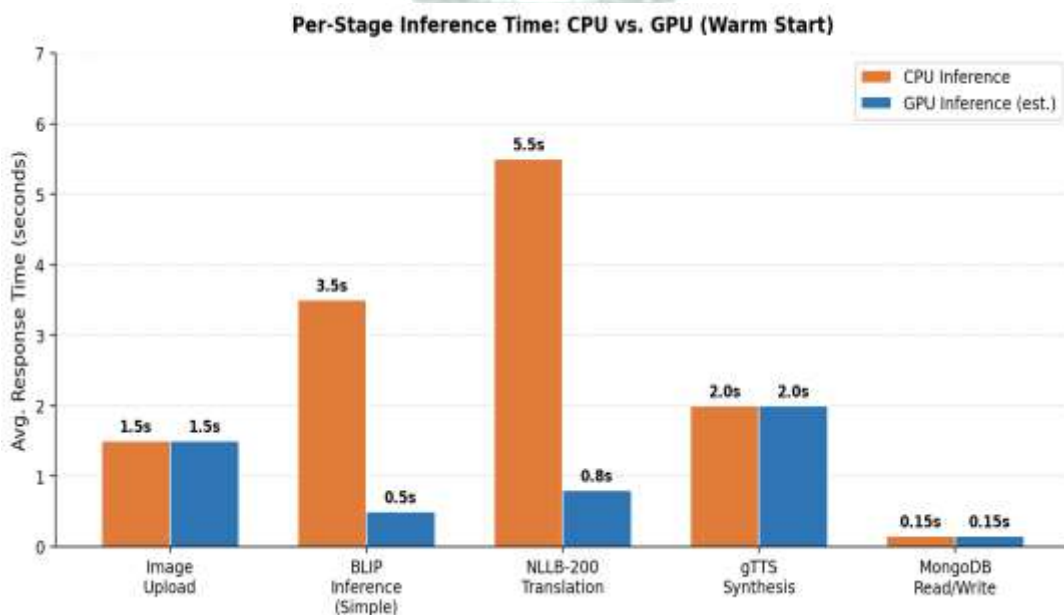
BLEU score comparison: NLLB-200 (600M distilled) vs. Helsinki-NLP MarianMT across eight representative language pairs. NLLB-200 yields the largest gains in low-resource South Asian languages (Telugu +94.4%, Tamil +69.0%)

**TABLE IV. NLLB-200 vs. MarianMT — Comparative Summary**

Criterion	MarianMT	NLLB-200 (ours)
Language coverage	Per-pair (~50 languages)	200 languages (single model)
Telugu quality	Repetition / hallucination	Coherent, structurally correct
Hindi quality	Acceptable, inconsistent	High quality
Inference memory	~500 MB per model	~2.4 GB (single shared model)
Deployment complexity	Multiple model files	Single unified model

### C. System Performance Benchmarking

Per-stage latency under CPU and estimated GPU conditions shows that cold-start latency (BLIP-Large: 25–40s; NLLB-200: 15–25s) affects first-request user experience. Warm-start inference (BLIP: 2–5s; NLLB: 3–8s; gTTS: 1–3s) is acceptable for interactive web use. All CPU timings measured on Intel Core i7-10th Gen, 16 GB RAM; GPU estimates assume NVIDIA T4 with CUDA 12.1.



Per-stage inference latency: CPU vs. estimated GPU (warm start). BLIP and NLLB-200 benefit most from CUDA acceleration (5–10× speedup). Upload, gTTS, and MongoDB latency are network-bound and GPU-independent.

## D. Security Analysis

bcrypt's 12 rounds yield ~0.3s per hash verification, making brute-force enumeration computationally infeasible. JWT tokens use HS256 signing with 24-hour expiry and iat claim. OTP codes expire in 5 minutes and are cleared from MongoDB after successful verification, closing the replay attack vector. A known limitation—absence of JWT validation on AI inference endpoints—is documented in Section VI.



Fig. 8. SceneSense AI

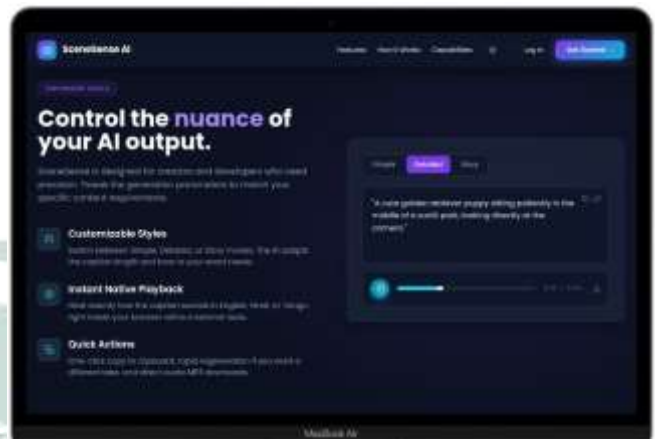


Fig. 9. SceneSense AI

## E. User Interface and Application Walkthrough

This section presents a complete visual walkthrough of the SceneSense AI interface across both the public landing layer and the authenticated application workspace. The UI is implemented in React 18 with a dark-first glassmorphism design system and Framer Motion animations. All screenshots were captured from the running localhost deployment.

The workspace interface demonstrates the direct mapping between UI controls and backend model parameters: the Caption Style selector maps to BLIP generation mode, the Target Language dropdown maps to the NLLB-200 BCP-47 forced BOS token, and the Voice Assistant toggle maps to the gTTS VoiceService call. This design ensures the system's internal AI pipeline is transparently surfaced to the end user.

## VI. ADVANTAGES AND LIMITATIONS

### A. Key Advantages

- Comprehensive multilingual support: 20 languages via a single NLLB-200 model eliminates per-language-pair model management, particularly for South Asian and East Asian languages underserved by existing tools.
- Accessibility through voice: Integrated gTTS synthesis extends image understanding to visually impaired users and low-literacy populations, aligning with WCAG 2.1 accessibility principles.
- Three-mode caption flexibility: Simple, Detailed, and Story modes address diverse downstream use cases—from alt-text generation to creative content production—within a single application.
- Resource-efficient serving: Singleton lazy-loading ensures both BLIP and NLLB-200 are loaded once per server session, making the system viable on commodity hardware without model-serving infrastructure.
- Zero per-request cost: Fully open-source and self-hostable, unlike Azure Vision or Google Cloud Vision APIs, making it economically viable for educational institutions and NGOs.

### B. Limitations

- Memory footprint: BLIP-Large (~900 MB) and NLLB-200 (~2.4 GB) running concurrently require ~3.5 GB RAM, exceeding free-tier server allocations (512 MB). Quantization or model offloading is required for budget hosting.

- Cold-start latency: Combined first-request model loading time of 40–65 seconds creates a poor initial user experience after server restart.
- Internet dependency: gTTS requires server-side internet connectivity; network outages cause voice generation failure.
- Missing per-user data isolation: The history collection is not segmented by authenticated user, creating a privacy concern in multi-user deployments.
- Unauthenticated AI endpoints: /upload, /caption, /translate, /voice, and /history do not require a valid JWT token, representing a security gap for production deployments.

## VII. FUTURE SCOPE

- Per-user data isolation: Link history records to authenticated users via a user\_id field and enforce JWT validation on all AI inference endpoints.
- Model quantization: Apply 8-bit or 4-bit quantization (via HuggingFace bitsandbytes) to reduce memory consumption by 50–75%.
- Asynchronous task queue: Integrate Celery + Redis for background AI inference with WebSocket result notification.
- Offline TTS: Replace gTTS with a locally-hosted neural TTS model (Coqui TTS) to eliminate internet dependency.
- Real-time video captioning: Extend pipeline to video streams using OpenCV frame extraction with BLIP per-frame captioning.
- Cloud object storage: Migrate file storage to Amazon S3 or Google Cloud Storage for horizontal scaling.
- WCAG 2.1 AA compliance: Add full ARIA labels, keyboard navigation, and screen reader optimization.
- Domain-specific fine-tuning: Fine-tune BLIP on medical imaging, satellite imagery, or e-commerce product photography datasets.

## VIII. CONCLUSION

This paper has presented SceneSense AI, a full-stack AI-powered web application integrating image captioning, multilingual translation, and voice synthesis. The system leverages Salesforce BLIP-Large (CIDEr: 133.3) for high-quality English caption generation across three contextual modes, Facebook NLLB-200 (600M distilled) for translation into twenty global languages, and Google gTTS for text-to-speech output. The authenticated workspace exposes these capabilities through a polished React 18 interface with direct user control over model parameters.

Benchmark evaluation demonstrates acceptable warm-start inference latency on CPU hardware with significant headroom under GPU acceleration. SceneSense AI provides zero-per-request-cost, self-hosted, verified support for regional Indian languages alongside 16 other global languages. As a Final Year B.Tech project, it constitutes a comprehensive case study in deep learning model serving, RESTful API design, NoSQL data persistence, modern frontend engineering, and application security—demonstrating that high-impact, accessibility-focused AI tools can be built and deployed by small teams using open-source



technologies.

Fig. 10. SceneSense AI

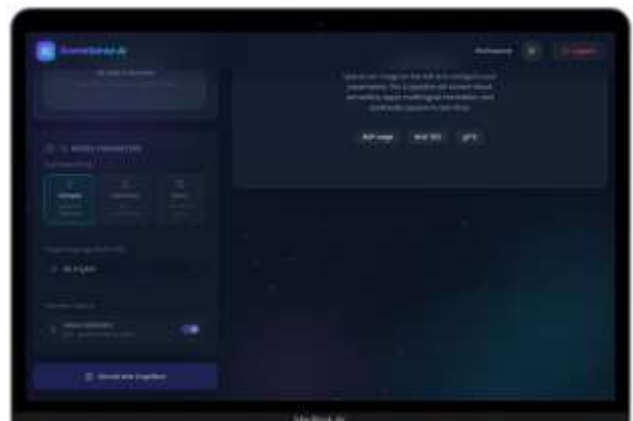


Fig. 11. SceneSense AI

## ACKNOWLEDGEMENTS

We sincerely thank the Department of Computer Science Engineering, Krishna University, Machilipatnam, for providing the necessary infrastructure and support throughout this project. We also acknowledge Salesforce Research for BLIP, Meta AI Research for NLLB-200, and the Hugging Face Transformers team for making their pre-trained models openly available. This project was completed as part of our final-year B.Tech under the guidance of our faculty advisors at Krishna University.

## REFERENCES

- [1]Internet Live Stats, "Number of photos taken per day," 2024. [Online]. Available: <https://www.internetlivestats.com>
- [2]A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
- [3]Ethnologue, "Languages of the World," 25th ed., SIL International, 2022.
- [4]C. Fang et al., "From captions to visual concepts and back," in *Proc. IEEE CVPR*, 2015, pp. 1473–1482.
- [5]Y. Yang et al., "Corpus-guided sentence generation of natural images," in *Proc. EMNLP*, 2011, pp. 444–454.
- [6]O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE CVPR*, 2015, pp. 3156–3164.
- [7]K. Xu et al., "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. ICML*, 2015, pp. 2048–2057.
- [8]X. Li et al., "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Proc. ECCV*, 2020, pp. 121–137.
- [9]P. Zhang et al., "VinVL: Revisiting visual representations in vision-language models," in *Proc. IEEE CVPR*, 2021, pp. 5579–5588.
- [10]J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training," in *Proc. ICML*, 2022, pp. 12888–12900.
- [11]P. Koehn et al., "Moses: Open source toolkit for statistical machine translation," in *Proc. ACL*, 2007, pp. 177–180.
- [12]I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NeurIPS*, vol. 27, 2014.
- [13]J. Tiedemann and S. Thottingal, "OPUS-MT — Building open translation services for the world," in *Proc. EAMT*, 2020.
- [14]NLLB Team, Meta AI Research, "No language left behind: Scaling human-centered machine translation," *arXiv:2207.04672*, 2022.
- [15]A. van den Oord et al., "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.
- [16]J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. IEEE ICASSP*, 2018, pp. 4779–4783.
- [17]T. Wolf et al., "Transformers: State-of-the-art natural language processing," in *Proc. EMNLP: System Demonstrations*, 2020, pp. 38–45.
- [18]MongoDB, Inc., "MongoDB Atlas Documentation," 2024. [Online]. Available: <https://www.mongodb.com/docs/atlas/>
- [19]A. Radford et al., "Language models are unsupervised multitask learners," *OpenAI Blog*, 2019. (GPT-2)
- [20]M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," *RFC 7519*, IETF, 2015.
- [21]N. Provos and D. Mazieres, "A future-adaptable password scheme," in *Proc. USENIX ATC*, 1999. (bcrypt)
- [22]gTTS Contributors, "gTTS: Google Text-to-Speech Python library," 2024. [Online]. Available: <https://gtts.readthedocs.io>
- [23]Pallets Projects, "Flask Documentation, v3.0," 2024. [Online]. Available: <https://flask.palletsprojects.com>
- [24]Meta AI, "PyTorch Documentation," 2024. [Online]. Available: <https://pytorch.org/docs/>
- [25]W3C, "Web Content Accessibility Guidelines (WCAG) 2.1," 2018. [Online]. Available: <https://www.w3.org/TR/WCAG21/>