Survey Report on VisionStack AI Studio

Onkar Phopase[1], M. K. Nivangune[2], Sujal Gangarde[3] , Kunal Bobade[4], Vaibhav Pisute[5]

Department of Computer Engineering, Sinhgad Academy of Engineering, Pune

---***---

ABSTRACT

Recent advancements in artificial intelligence have positioned computer vision as one of the most influential technologies in the modern digital landscape. From healthcare diagnostics to autonomous vehicles and smart surveillance, the ability of machines to perceive and interpret visual data has transformed automation. However, developing such vision-based systems remains complex due to fragmented tools, extensive coding requirements, and high computational costs.

Vision Stack AI Studio aims to overcome these challenges by providing an integrated, modular platform for designing, training, and deploying visual intelligence models. It combines deep learning techniques particularly convolutional neural networks (CNNs) with an intuitive interface that streamlines dataset management, model evaluation, and deployment. The system enables both developers and non-technical users to build robust vision models for classification, detection, and real-time prediction with minimal effort.

***Key Words*:** Computer Vision, Artificial Intelligence, Deep Learning, CNN, Image Recognition, AI Automation, Model Deployment.

1.INTRODUCTION

Artificial Intelligence (AI) has become central to modern technology, with computer vision being one of its fastest-growing domains. The ability to process and interpret images or videos allows machines to replicate human perception, enabling applications such as medical image diagnosis, industrial defect detection, and autonomous navigation. Despite the availability of advanced frameworks like TensorFlow, PyTorch, and OpenCV, building a fully functional computer vision pipeline still demands significant technical expertise. Users must handle dataset preprocessing, model architecture design, hyperparameter optimization, and deployment manually making the process time-consuming and error-prone.

Vision Stack AI Studio addresses this gap by offering a unified ecosystem that automates the entire computer vision workflow. It allows users to import data, train deep learning models, visualize performance metrics, and deploy models through APIs or local environments. The platform is designed to promote accessibility, efficiency, and scalability, making AI-based vision solutions feasible for a wider range of users and industries.

2. LITERATURE RIVIEW

2.1 Deep Learning in Computer Vision

Deep learning has revolutionized image recognition and visual analytics. CNNs, first popularized through **AlexNet** (Krizhevsky et al., 2012), demonstrated the power of hierarchical feature extraction in large-scale image datasets. Later models such as **VGGNet** (Simonyan & Zisserman, 2014) and **ResNet** (He et al., 2016) introduced deeper architectures and residual learning, improving both speed and accuracy.
These advancements form the foundation of Vision Stack AI Studio, which integrates CNN models for automated image classification and object detection while allowing custom model fine-tuning.

2.2 Existing Platforms

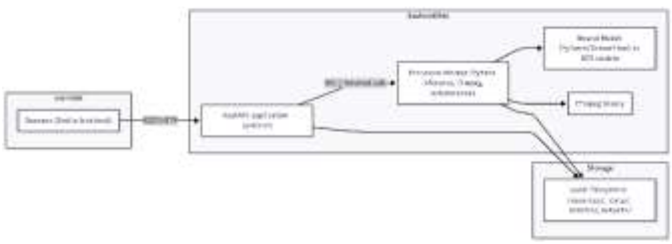
Several platforms provide partial automation for visual AI tasks. **RoboFlow**, **Google AutoML Vision**, and **CVAT** assist with data labeling and training but lack comprehensive features for deployment and workflow integration. Research on modular vision platforms emphasizes the need for unified interfaces that handle the complete model lifecycle from data ingestion to API creation.

2.3 Research Gap

Although numerous tools exist for specific vision tasks, few solutions offer **end-to-end automation** within a single system. Many require manual setup, complex configuration, and fragmented toolchains. **VisionStack AI Studio** bridges this research gap by combining data preprocessing, model training, visualization, and deployment into a cohesive framework supported by modern AI techniques.

3. PROPOSED SYSTEM ARCHITECTURE

The proposed system is designed as a layered architecture integrating data management, machine learning, and deployment components.

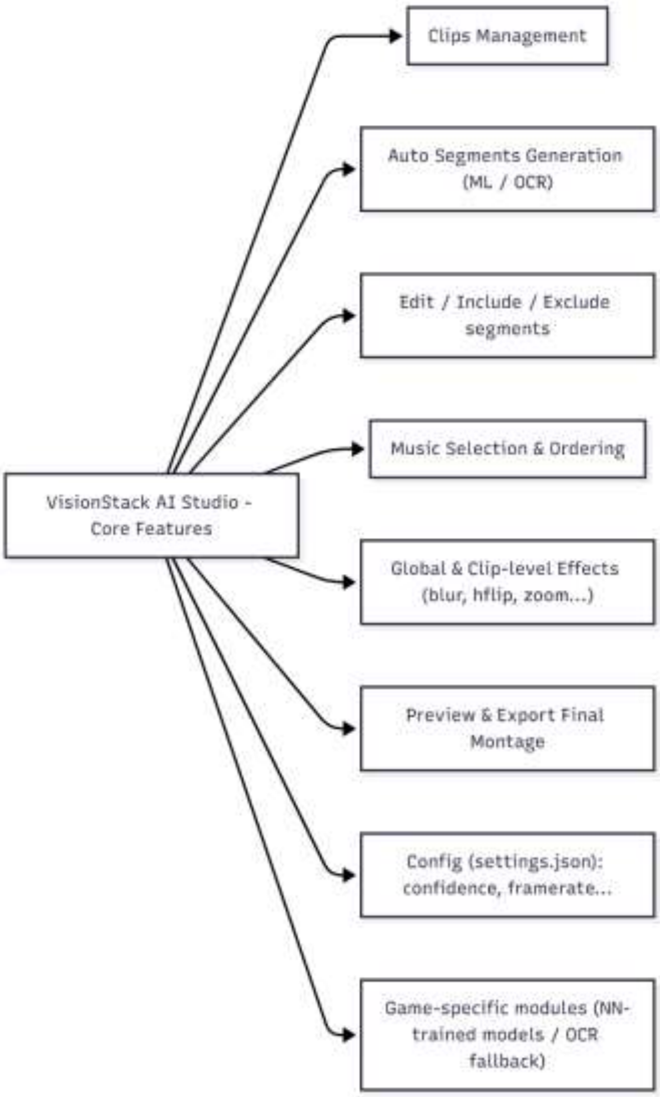


System Architecture Diagram

3.1 System Components

1. **Dataset Manager:** Handles image uploads, annotations, resizing, normalization, and augmentation.
2. **AI Engine:** Incorporates CNN architectures for feature extraction and classification, supporting both pre-trained and custom models.
3. **Evaluation Module:** Monitors metrics such as accuracy, precision, recall, and F1-score to assess model performance.
4. **Deployment Layer:** Exports trained models as REST APIs or local inference services for integration with web or mobile applications.
5. **Visualization Dashboard:** Displays training progress, confusion matrices, and real-time prediction outputs.

3.2 System Features

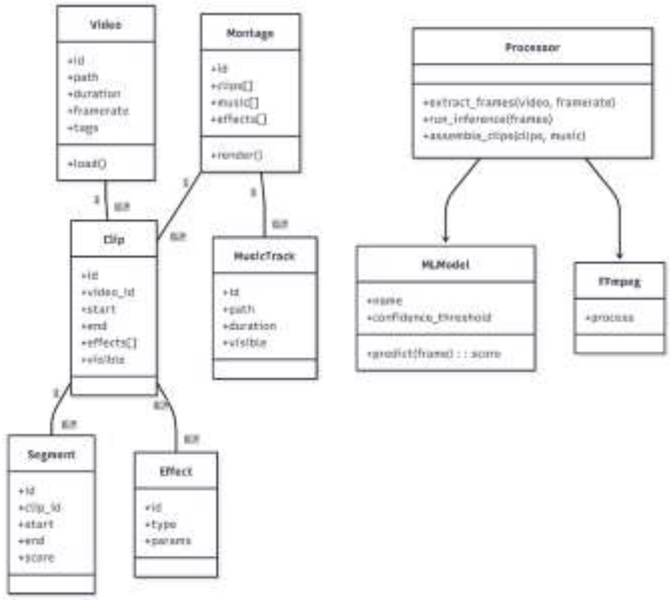


System Feature Diagram

- User-friendly web interface for non-programmers.
- Support for multiple deep learning frameworks (TensorFlow, PyTorch).
- Real-time inference and performance tracking.
- Scalable deployment options for edge or cloud environments.

4. METHODOLOGY

The workflow of **Vision Stack AI Studio** follows a structured pipeline that ensures reliability, efficiency, and reproducibility in visual model creation.



UML Class Diagram

1. **Data Acquisition:** Users upload labeled or unlabeled image datasets in supported formats.
2. **Preprocessing:** The system automatically resizes, normalizes, and augments images to enhance model generalization.
3. **Model Selection:** Users select a model architecture such as ResNet, EfficientNet, or MobileNet based on use case requirements.
4. **Training Process:** The training module applies forward propagation, computes loss using categorical cross-entropy, and optimizes parameters via backpropagation using stochastic gradient descent (SGD) or Adam optimizer.

UML Sequence

1. **Evaluation:** The trained model is validated using test data, and metrics such as accuracy, recall, and precision are calculated.
2. **Deployment:** The finalized model is exported through an API or hosted service, enabling real-time predictions.

3. **Feedback Loop:** Users can analyze performance and retrain models to enhance accuracy over time.

5. TECHNOLOGY STACK

Layer	Technology / Tool
Frontend	React.js, Tailwind CSS
Backend	Flask / FastAPI
AI Libraries	TensorFlow, PyTorch, OpenCV
Database	MongoDB, SQLite
Deployment	Docker, RESTful APIs
Hardware	NVIDIA GPU, Intel i5 or higher, 8 GB+ RAM

This combination ensures both usability and scalability for local and cloud-based AI operations.

6. CONCLUSION

Vision Stack AI Studio introduces a practical approach to simplifying complex computer vision workflows. The system integrates dataset processing, deep learning model training, and API-based deployment into a single cohesive platform. Its modular design allows for flexibility in architecture selection, while the user-friendly interface minimizes dependency on advanced coding knowledge.

Future Enhancements

1. Integration with **Edge AI** for low-latency inference on IoT devices.
2. Incorporation of **Explainable AI (XAI)** modules to interpret model predictions.
3. Support for **video analytics** and real-time object tracking.
4. Expansion into **cloud-native deployment** using services like AWS, Azure, and Google Cloud.
5. Automated hyperparameter tuning using AutoML techniques.

By advancing accessibility and automation in visual intelligence, Vision Stack AI Studio can accelerate innovation across multiple industries where visual data plays a critical role.

7. ACKNOWLEDGMENT

The authors would like to express their gratitude to **Prof. Mr. M. K. Nivangune** for his continuous guidance, insightful feedback, and encouragement throughout the development of this project. The team also extends sincere thanks to the **Department of Computer Engineering, Sinhgad Academy of Engineering, Pune**, for providing the facilities and technical support required to carry out this research successfully. Finally, the authors acknowledge their peers and families for their motivation and support during the completion of this work.

8. REFERNCES

1. P. Shukla, H. Sadana, A. Bansal, D. Verma, C. Elmadjian, B. Raman, and M. Turk, “Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Salt Lake City, UT, USA, 2023, pp. 1–10.

2. C. Midoglu, S. S. Sabet, M. H. Sarkhoosh, M. Majidi, S. Gautam, H. M. Solberg, T. Kupka, and P. Halvorsen, “AI-Based Sports Highlight Generation for Social Media,” in Mile-High Video Conference (MHV ’24), Denver, CO, USA, 2024, pp. 1–7, doi: 10.1145/3638036.3640799.

3. V. Edithal, L. Zhang, I. Blank, and I. Junejo, “Gameplay Highlights Generation,” arXiv preprint arXiv:2505.07721, 2025

4. C. Ringer and M. A. Nicolaou, “Deep Unsupervised Multi-View Detection of Video Game Stream Highlights,” in Proceedings of the 13th International Conference on Foundations of Digital Games (FDG), 2018, pp. 1-6, doi: 10.1145/3235765.3235781.