IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

Viraasat: A Human-in-the-Loop Platform for Digitizing, Transcribing, Modernizing and Translating Historical Marathi Manuscripts

Prof. Shafique Shaikh, Prof. Sangita Chaudhari, Pranav Bhanudas Mane
Anurag Bapusaheb Wable, Aditya Bhausaheb Aher
Project Guide, Dept. of Information Technology
Students, Dept. of Information Technology
Sandip Institute of Technology and Research Centre, Nashik

Abstract: *Marathi manuscripts such as royal orders, land grants, Modi-script records and medieval literature are deteriorating, and only a few specialists can read them. Scanning preserves them as images but does not make them searchable, and general-purpose OCR tools fail on faded ink, skewed pages, archaic orthography and Modi script. This paper presents the design and interim progress of Viraasat (Marathi for "heritage"), a full-stack web platform that takes a scan or photograph of a historical Marathi manuscript through a nine-stage workflow: ingestion, image preprocessing, text-line segmentation, AI-assisted transcription, human review, modernization of archaic Marathi, translation into Hindi and English, synchronized reading and export. Preprocessing combines deskewing, CLAHE contrast enhancement and edge-preserving filtering. Lines are located by signal processing on the Devanagari headline (shirorekha) using projection profiles, smoothing and peak and valley detection. Transcription is delegated to interchangeable providers (Google Gemini multimodal vision, Hugging Face Indic TrOCR and Tesseract), and doubtful words are bracketed with low confidence. A scholar reviews every line, every correction is stored in an audit ledger, and the original scan and raw OCR are never changed. The platform follows five archival design principles and is built as a three-tier system with a React front end, a Node.js back end, a Python AI service and MongoDB. Progress recorded up to October 2026 (about 40% of the project) comprises the design, preprocessing, line segmentation, OCR providers, back end, review interface and automated tests. Evaluation of recognition accuracy, validation of modernization and translation, and user testing remain pending, and no performance figures are claimed. The system is a research prototype, and its output requires scholarly verification..*

Key Words: historical document analysis, Marathi OCR, Modi script, Devanagari, shirorekha, human-in-the-loop, multimodal models, digital heritage

1. INTRODUCTION

India holds millions of manuscripts in archives, temples and private collections. Marathi alone has a rich record of 17th-century royal orders from the era of Chhatrapati Shivaji Maharaj, Peshwa-era land grants (sanad and mahajar), Modi-script administrative papers and medieval literature such as the Dnyaneshwari. These documents are valuable to historians, linguists and the public, but reading them is hard for four reasons.

1 Physical decay. Paper and parchment are stained, torn, faded, discolored and often photographed at a slant.

2 Script and orthography. The writing uses Modi script or archaic Devanagari with continuous headlines, ligatures and obsolete spellings.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40532

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

3 Archaic grammar and vocabulary. Case endings and Persian or Sanskrit administrative terms from the 13th to 17th centuries are not understood by ordinary readers today.
4 Inadequate general OCR. Off-the-shelf tools trained on printed text either fail or produce confident but wrong output on manuscripts.

Digitization therefore needs more than scanning. It needs cleaning of the image, finding the lines, reading the text, checking the reading, translating the language into something a modern reader understands, and keeping the link between each line of text and its place on the original page. Viraasat is designed to provide this whole chain in one platform, and to do it in a way that a scholar can trust and audit.

The contributions of this paper are:

• A nine-stage, end-to-end workflow for historical Marathi manuscripts, from image upload to export.
• A set of five archival design principles that keep the original evidence intact and make every human correction traceable.
• A line segmentation method that exploits the shirorekha of Devanagari text using projection profiles and peak analysis.
• A provider-based OCR design that allows multimodal LLM, transformer and classical engines to be swapped without changing the rest of the system.
• A synchronized multilingual reader that aligns the scan, raw OCR, modern Marathi, Hindi and English line by line.

II. LITERATURE REVIEW

Research on historical document recognition, image preprocessing and multimodal models was reviewed. The main findings are grouped below.

II.1 OCR engines

Tesseract [1] is the most widely used open-source OCR engine and supports Devanagari, but it is trained largely on printed text and degrades on handwriting and aged paper. Line-level sequence recognizers such as the CRNN with CTC loss [2], [3] learn to read whole line images without character segmentation and are well suited to connected scripts. Transformer-based recognizers such as TrOCR [4] pair an image encoder with a text decoder and have been extended to several languages, including Indic scripts.

II.2 Multimodal models

General-purpose multimodal models such as Gemini [5] can read text in images and follow instructions in a prompt, which makes it possible to ask for a particular transcription style and output format. They are not trained specifically for Modi or Shivakalin handwriting, and their output can vary and can include text that is not on the page, so their results need verification.

II.3 Image preprocessing

Global thresholding [6] fails on unevenly stained pages. Contrast-limited adaptive histogram equalization (CLAHE) [7] improves local contrast while limiting noise amplification, and bilateral filtering [8] smooths noise while keeping edges, which matters for thin Devanagari marks.

II.4 Layout analysis

Projection-profile methods find text lines by summing ink along rows. They are simple and fast, and for Devanagari they exploit the headline that joins the letters of a word.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

II.5 Research Gap

Existing tools address individual steps. We found no openly documented system that combines image cleaning, line segmentation, historical transcription, expert review with an audit trail, modernization of archaic Marathi, multilingual translation and line-level provenance in one workflow for Marathi manuscripts.

III. DESIGN PRINCIPLES

Because the system handles historical evidence, five principles guide the design.

1 Immutable scans. The uploaded image is never modified or overwritten. Preprocessed copies are stored beside it.

2 Immutable raw OCR. The machine's original reading is kept permanently in a field called ocrText. Human edits go in a separate correctedText field.

3 Honest uncertainty. Faded, torn or ambiguous words are put in square brackets, for example [unclear] in Marathi, with low confidence scores, and lines with confidence below 0.75 are flagged as uncertain.

4 Human-in-the-loop audit trail. Every correction, verification or deletion is recorded in a correction-history ledger, and the automated pipeline stops after OCR until a scholar has reviewed the text.

5 Synchronized provenance. Selecting a line in modern Marathi, Hindi or English highlights the exact region of the original scan.

IV. SYSTEM ARCHITECTURE

The platform has three tiers connected by REST over HTTP, plus a database and optional job queue.

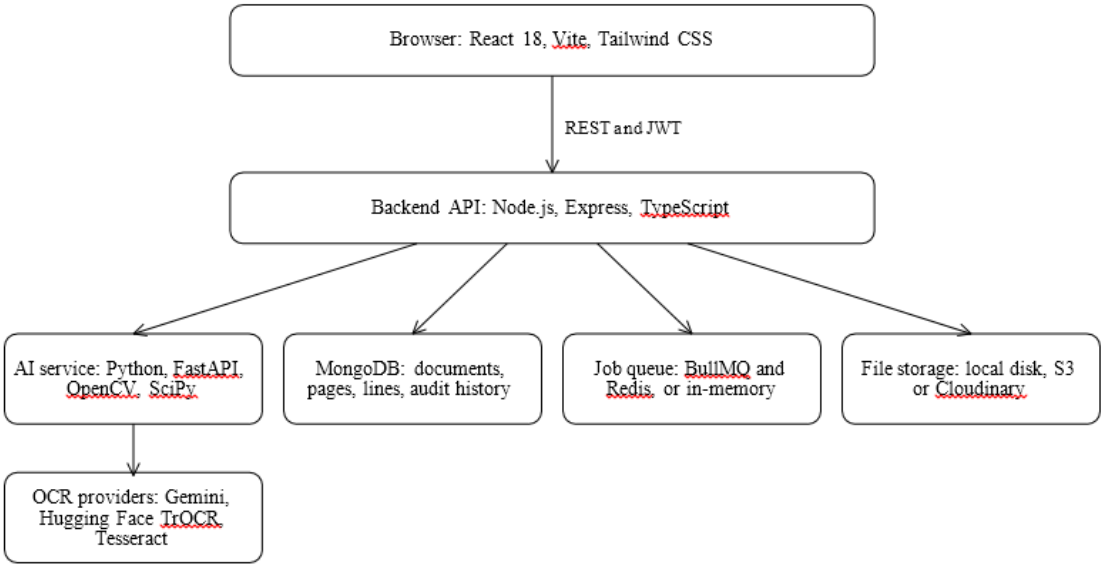


Fig. 1. Architecture of Viraasat.

Front end. A React 18 single-page application built with Vite and styled with Tailwind CSS. It provides upload, a dual-pane review canvas with bounding boxes, a progress tracker, a synchronized multilingual reader and an export dialog. TanStack Query handles server state and polling of job status.

Back end. A Node.js and Express API written in TypeScript. It handles JWT authentication with bcrypt password hashing, rate limiting, file upload through Multer, document and line management, the processing queue, and PDF, JSON and TXT export. Long-running work is queued, so HTTP requests return immediately with a job identifier. BullMQ with Redis is used in production, with an in-memory fallback for local use.

AI service. A Python FastAPI service with endpoints for preprocessing, segmentation, OCR, modernization and translation. It uses Pillow, NumPy, SciPy and OpenCV.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

Data layer. MongoDB holds users, documents, pages, text lines, translations and processing jobs. A document moves through a state machine: uploaded, preprocessing, segmenting, OCR processing, OCR review, modernizing, translating and completed, with a failed state reachable from any stage.

V. METHODOLOGY

V.1 Workflow

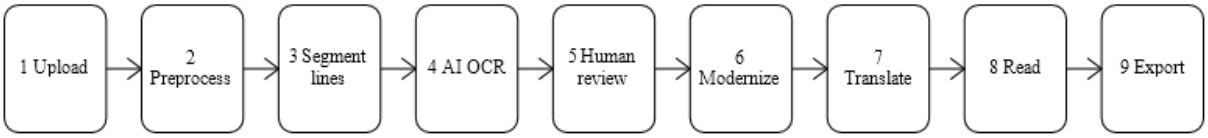


Fig. 2. End-to-end workflow. The pipeline pauses at stage 5 until a scholar finalizes the text.

The user uploads a JPG, PNG or WEBP image or a multi-page PDF, which is split into page images, or chooses a bundled sample manuscript. The backend stores the files, creates records, and queues a job. The stages run in order, with progress reported to the interface.

V.2 Image Preprocessing

The preprocessing stage prepares a page for segmentation without destroying fine ink strokes.

Deskewing. Ink pixels isolated by an Otsu threshold form a point set. The minimum-area enclosing rectangle of this set gives a rotation angle theta, and deskewing is applied when the absolute angle lies within a configured range (0.4 to 25 degrees). The image is rotated about its center with bicubic interpolation to avoid breaking fine strokes such as kana and matra marks.

Illumination and contrast. The image is converted to the CIELAB color space and CLAHE is applied to the lightness channel over a grid of 8 x 8 tiles. For each tile the histogram h(i) is clipped at a limit beta, and the excess is redistributed across the bins:

$$h_{clipped}(i) = \min(h(i),\ \beta) \tag{1}$$

Tile borders are blended by bilinear interpolation to avoid visible seams. The configured clip limit is beta = 2.0.

Noise reduction. Edge-preserving filtering is used because a Gaussian blur damages thin Devanagari marks such as the velanti, anusvara and virama. A bilateral filter weights neighboring pixels by both spatial distance and intensity similarity [8]:

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) \exp\left(-\frac{\|x - x_i\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{\|I(x) - I(x_i)\|^2}{2\sigma_r^2}\right) \tag{2}$$

The design uses sigma_s = sigma_r = 35. A median filter is available as a simpler alternative. Scanner borders are cropped, and a list of the operations applied is stored for each page.

V.3 Line Segmentation Using the Shirorekha

In Devanagari and Modi manuscripts the words of a line hang from a horizontal headline, so each line produces a strong horizontal band of ink. The method uses this property.

1 Inversion and normalization. The grayscale image is inverted so that ink becomes positive and then scaled to [0, 1].

2 Column margins. A vertical projection C(x) is computed, and columns with values above a threshold of 0.35 times the mean mark the left and right edges of the written column, which excludes empty borders.

3 Row profile. Within the column, the horizontal projection is:

$$R(y) = \sum_{x = x_{left}}^{x_{right}} \tilde{I}(x, y) \tag{3}$$

4 Smoothing. The profile is convolved with a moving-average box kernel K whose size is proportional to the image height (height divided by 80):

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

$$S(y) = (R * K)(y) = \sum_{\tau} R(\tau)\, K(y - \tau) \quad (4)$$

5 Headline peaks. Local maxima p1, p2, ..., pk of S(y) correspond to the headlines. They are found with scipy.signal.find_peaks using a minimum distance of H/45 and a prominence of at least 3.5% of the maximum of S, where H is the image height. This suppresses sub-line strokes and background noise.

6 Valley splitting. The boundary between line i and line i + 1 is the point of minimum ink density between their peaks:

$$v_i = \arg \min_{y \in [p_i,\, p_{i+1}]} S(y) \quad (5)$$

7 Bounding boxes. Each interval between valleys gives a box [x_left, v(i-1), x_right - x_left, v(i) - v(i-1)]. The line is cropped and stored with its coordinates, which later enable synchronized highlighting.

V.4 Transcription

Transcription is performed by an OCR provider chosen by configuration, behind a common interface that returns, for each line, its text, a confidence score and word-level confidences.

Gemini (multimodal vision). The whole page image is sent with a structured prompt. The prompt asks the model to act as an expert in historical Marathi handwriting (Modi script, archaic Devanagari, Shivakalin and Peshwa-era writing), tells it the exact number of lines found by segmentation, and requires it to keep the historical orthography, to place faded or doubtful characters in square brackets, and to give low confidence (about 0.40 to 0.70) to ambiguous words and high confidence (about 0.90 to 0.99) to legible ones. The model is called in JSON mode with a low temperature (0.1), so the output can be parsed against a fixed schema and matched to the segmented line boxes.

Hugging Face Indic TrOCR. A transformer recognizer with an image encoder and a text decoder performs best on single text lines. The provider therefore crops each segmented line, sends it to the model, and merges the results, giving line-level confidence to the word tokens.

Tesseract. A local Devanagari engine is available for offline use.

Development provider. For offline development and interface testing, the system includes a simulated provider that returns predefined texts for sample manuscripts. It exists only to exercise the pipeline and user interface, and its output is not a recognition result.

The raw output is stored in ocrText, an editable copy is stored in correctedText, and the document status moves to OCR review.

V.5 Human Review

The review interface has two panes. The left pane shows the page with highlighted line boxes, zoom and a toggle between original and preprocessed images. The right pane lists the lines with word-level confidence shown by color: green above 0.85, yellow from 0.70 to 0.85 and red below 0.70. The scholar can edit words, accept, reject or merge lines, tag damaged characters and mark a line as verified. Every edit writes an entry to the correction history without changing ocrText. The scholar then finalizes the text to start modernization.

V.6 Modernization

Archaic Marathi is converted to contemporary Marathi in a way that is safe for historical content. A rule-based engine with more than 120 substitution rules covers particles, verb forms, case endings, locatives, names of weekdays and spelling variants. Examples include the archaic particle aaji becoming aaj, jaahali becoming jhaali, asonu becoming asoon, varooni becoming varoon, and the dative ending si becoming laa. Proper names, fort names, royal titles, dates and bracketed uncertainties are protected and are not altered. An LLM-based mode can be used instead. The rules applied are saved as a change list, so every modernization can be inspected. The engine also produces a short explanatory commentary on the document's historical context.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

V.7 Translation and Synchronized Reading

Modern Marathi is translated into Hindi and English, line by line, so that the number of lines is preserved. The synchronized reader shows, for each line, the historical text, modern Marathi, Hindi and English. Clicking a line highlights its bounding box on the scan, and a language toggle shows the explanatory commentary in Marathi, Hindi or English. Results can be exported as a formatted PDF report, a JSON file with boxes, confidences and the correction history, or plain text.

VI. IMPLEMENTATION

Table 1. Technologies used in Viraasat.

Layer	Technologies
Front end	React 18, TypeScript, Vite, Tailwind CSS, TanStack Query, Axios, React Hook Form, Zod, Framer Motion
Back end	Node.js, Express, TypeScript, Mongoose, JWT, bcryptjs, Multer, Helmet, pdf-lib, BullMQ with Redis
AI service	Python 3.11, FastAPI, Uvicorn, Pillow, NumPy, SciPy, OpenCV
OCR providers	Google Gemini multimodal vision, Hugging Face Indic TrOCR, Tesseract
Data	MongoDB, with collections for users, documents, pages, text lines, translations, correction history and processing jobs
Deployment	Docker and Docker Compose, with Redis, S3 or Cloudinary storage optional

VII. TESTING AND EVALUATION PLAN

Software testing. Each tier has automated tests. The AI service uses pytest to check that segmentation detects lines on a real manuscript image, that OCR returns the expected number of lines, that the modernization and translation endpoints return their required fields, and that JSON serialization is safe. The back end uses Jest and Supertest for authentication, document operations, line verification and the modernization and translation contracts. The front end uses Vitest and Testing Library for components, forms and API mocks.

Recognition and language quality. The tests above check functional behavior, not recognition accuracy. No quantitative accuracy has been measured yet. The planned evaluation is summarized below.

Table 2. Planned evaluation of each component.

Component	Metric	Method	Status
Preprocessing	Visual quality and effect on OCR error	Compare OCR error with and without each step	Planned
Line segmentation	Line detection rate and box overlap (IoU)	Compare with manually drawn line boxes	Planned
Transcription	Character Error Rate (CER) and Word Error Rate (WER)	Compare each provider with expert transcriptions on a held-out set	Planned
Uncertainty marking	Whether low-confidence words are the wrong ones	Compare confidence with correctness	Planned
Human review	Time and number of corrections per page	Timed sessions with scholars	Planned
Modernization	Rule precision and preserved entities	Expert check of changed and protected words	Planned
Translation	Adequacy and fluency	Rating by bilingual reviewers	Planned

CER and WER are computed from edit distance against an expert transcription:

$$\mathrm{CER} = \frac{S + D + I}{N} \qquad (6)$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 1, November 2026
Impact Factor: 8.2

where S, D and I are the substitutions, deletions and insertions needed to turn the recognized text into the reference, and N is the number of reference characters. WER is computed in the same way on words.

VIII. PROGRESS AND CURRENT STATUS

The project is in progress, and about 40% of the work was complete at the time of writing. Table 3 summarizes the status of each component. The paper makes no accuracy claims, and quantitative results will be reported after the evaluation in Table 2 is carried out.

Table 3. Status of project components.

Component	Status
Design, architecture, data model and archival principles	Completed
Image preprocessing (deskew, CLAHE, bilateral filter)	Implemented; parameter tuning ongoing
Shirorekha line segmentation	Implemented; touching lines and margin notes need refinement
OCR providers (Gemini, Hugging Face TrOCR, Tesseract) and a simulated test provider	Implemented; accuracy not yet measured
Back end, job queue and database	Implemented
Front end review screen and correction audit trail	Implemented
Automated tests (pytest, Jest, Vitest) and Docker Compose	Written
Modernization, translation and export	First versions; validation pending
Accuracy evaluation (CER and WER)	Pending
Expert validation and user testing	Pending

IX. DISCUSSION AND LIMITATIONS

Strengths. The system treats manuscripts as evidence. Keeping the scan and the raw OCR unchanged, recording every correction and marking doubt openly let a scholar trust and audit the result. The provider-based OCR design lets the team switch to a better engine, or to a locally trained model, without touching the rest of the pipeline. Alignment of every line with its place on the scan makes the translations checkable.

Limitations.

• No accuracy figures yet. The recognition quality of each provider on Marathi manuscripts has not been measured, and claims about accuracy should wait for the evaluation above.

• Dependence on cloud models. Gemini and the Hugging Face API need internet access and an API key, may involve cost or usage limits, and send page images to a third party, which may matter for restricted archives. Their output can vary from run to run and may include text that is not on the page.

• Segmentation assumptions. The method relies on a strong headline per line. Curved or overlapping lines, marginal notes, and cursive Modi writing with weak headlines may need other methods.

• Rule-based modernization. The rules cover known forms but not every spelling variant, and a rule can be wrong in context, which is why entity protection and human review are needed.

• Translation quality. Machine translation of historical content can lose nuance and needs review for scholarly use.

• Scope. Testing so far used a small number of sample manuscripts.

X. CONCLUSION AND FUTURE WORK

This paper presented Viraasat, an end-to-end platform that helps turn images of historical Marathi manuscripts into searchable, verified and translated text while preserving the original evidence. Its main ideas are the five archival principles, shirorekha-based line segmentation, interchangeable OCR providers, a human-in-the-loop review with a full

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 1, November 2026 Impact Factor: 8.2

audit trail, and line-level synchronized reading. Future work includes building an annotated test set and measuring CER and WER for each provider, training or fine-tuning a local recognizer for Modi and archaic Devanagari to reduce dependence on cloud services, improving segmentation for curved and marginal text, expanding the modernization lexicon with expert input, evaluating translations with bilingual reviewers, and field testing with archivists and historians.

ACKNOWLEDGMENT

The authors thank their project guide and the Department of Information Technology, Sandip Institute of Technology and Research Centre, Nashik, for their guidance and support.

REFERENCES

[1] R. Smith, "An Overview of the Tesseract OCR Engine," in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR), 2007.
[2] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 39, no. 11, pp. 2298-2304, 2017.
[3] A. Graves, S. Fernandez, F. Gomez and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in Proc. ICML, 2006.
[4] M. Li et al., "TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models," arXiv:2109.10282, 2021.
[5] Gemini Team, Google, "Gemini 1.5: Unlocking Multimodal Understanding Across Millions of Tokens of Context," arXiv:2403.05530, 2024.
[6] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, 1979.
[7] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," in Graphics Gems IV, Academic Press, 1994, pp. 474-485.
[8] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 1998, pp. 839-846.
[9] P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," Nature Methods, vol. 17, pp. 261-272, 2020.
[10] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.