UnCodemy

Computer Vision Mini Project Ideas PDF

A computer vision mini project should use OpenCV and a webcam or a small dataset, take one to four weeks, and teach one technique such as thresholding, contour detection, face detection or optical flow. Document scanners, colour trackers, virtual paint apps and drowsiness alerts are good first projects.

Key takeaways

- Start with classic OpenCV methods before deep learning.
- A webcam is the only hardware you need for most ideas.
- Show each processing step on screen: it makes the viva easy.
- MediaPipe gives ready hand, face and pose landmarks for quick demos.
- Each idea fits in one to four weeks.

Who this PDF is for

This PDF is for 2nd and 3rd year students who want a hands-on computer vision mini project. Most ideas need only Python, OpenCV and a laptop webcam. They build skills in order: image basics, contours and colour spaces, landmarks, then small learned models.

How to choose your project

- **Webcam projects** demo well in a lab.
- **One technique per project** keeps it explainable.
- **Show intermediate images** (grayscale, edges, mask) in your report.
- **Use good lighting** when you test; note it as a limit.

Beginner mini projects (1 to 2 weeks)

1. Document Scanner

- **Problem:** Turn a phone photo of a page into a flat scan.
- **Dataset:** Your own photos of documents.
- **Tech stack:** Python, OpenCV.
- **Main modules:** Edge detection, contour finding, perspective transform, thresholding.
- **Steps:** Detect edges with Canny, find the largest four-point contour, warp to top view, threshold.
- **Show in the viva:** Each stage image from photo to scan.
- **Extensions:** Add OCR with Tesseract.

2. Colour Object Tracker

- **Problem:** Track a coloured ball in webcam video.
- **Dataset:** Live webcam.
- **Tech stack:** Python, OpenCV.
- **Main modules:** HSV conversion, colour mask, contour centre, path trail.
- **Steps:** Convert frames to HSV, mask the colour range, find the largest contour, draw its path.
- **Show in the viva:** Why HSV works better than RGB for colour.
- **Extensions:** Add trackbars to tune the colour range live.

3. Face and Eye Detection

- **Problem:** Detect faces and eyes in real time.
- **Dataset:** Live webcam; OpenCV's bundled Haar cascades.
- **Tech stack:** Python, OpenCV.
- **Main modules:** Cascade loading, detection, boxes, FPS counter.
- **Steps:** Load the cascades, detect faces, search for eyes inside face boxes, show FPS.
- **Show in the viva:** How Haar cascades work in simple terms.
- **Extensions:** Compare with a DNN face detector.

4. Virtual Paint with a Fingertip

- **Problem:** Draw on screen by moving a finger in the air.
- **Dataset:** Live webcam.
- **Tech stack:** Python, OpenCV, MediaPipe Hands.
- **Main modules:** Hand landmarks, fingertip tracking, canvas, colour picker.
- **Steps:** Get the index fingertip landmark, draw lines on a canvas, merge with the frame.
- **Show in the viva:** How you avoid drawing when the finger is lifted.
- **Extensions:** Add gesture-based eraser and save.

Intermediate mini projects (2 to 3 weeks)

5. Driver Drowsiness Alert

- **Problem:** Warn a driver when eyes stay closed too long.
- **Dataset:** Live webcam.
- **Tech stack:** Python, OpenCV, MediaPipe Face Mesh.
- **Main modules:** Eye landmarks, eye aspect ratio, timer, alarm.
- **Steps:** Compute eye aspect ratio each frame, start a timer when it drops, sound an alarm past a limit.
- **Show in the viva:** The eye aspect ratio formula and your threshold.
- **Extensions:** Add yawn detection.

6. Hand Gesture Volume Control

- **Problem:** Change system volume with a pinch gesture.
- **Dataset:** Live webcam.
- **Tech stack:** Python, MediaPipe Hands, an audio control library.

- **Main modules:** Landmark distance, smoothing, volume mapping.
- **Steps:** Measure thumb-to-index distance, smooth it, map it to the volume range.
- **Show in the viva:** How smoothing stops jitter.
- **Extensions:** Add play and pause gestures.

7. Printed Text Reader (OCR)

- **Problem:** Read printed text from photos of book pages, notices or signboards.
- **Dataset:** Your own photos.
- **Tech stack:** Python, OpenCV, Tesseract.
- **Main modules:** Preprocessing, text region detection, OCR, text clean-up.
- **Steps:** Grayscale, denoise and threshold, run Tesseract, compare results with and without preprocessing.
- **Show in the viva:** How preprocessing changed OCR quality.
- **Extensions:** Add Hindi with Tesseract's language data.

8. Coin Counter

- **Problem:** Count coins and total their value in a photo.
- **Dataset:** Your own photos of coins on a plain background.
- **Tech stack:** Python, OpenCV.
- **Main modules:** Blur, Hough circles, size-based value, total.
- **Steps:** Detect circles, group by radius, map sizes to values, sum.
- **Show in the viva:** Why camera height must stay fixed.
- **Extensions:** Train a small classifier for coin type.

Stretch mini projects (3 to 4 weeks)

9. Motion Detection Security Camera

- **Problem:** Record only when motion happens.
- **Dataset:** Live webcam or a phone used as IP camera.
- **Tech stack:** Python, OpenCV.
- **Main modules:** Background subtraction, motion area threshold, recording, snapshot alerts.
- **Steps:** Use frame differencing or MOG2, trigger recording above an area threshold, save clips.
- **Show in the viva:** How you avoided false alarms from light changes.
- **Extensions:** Send a snapshot to a Telegram bot.

10. Real-time Object Detection Demo

- **Problem:** Detect everyday objects with a pretrained model.
- **Dataset:** COCO-pretrained model weights; live webcam.
- **Tech stack:** Python, a YOLO model or OpenCV DNN with MobileNet-SSD.
- **Main modules:** Model loading, inference, box drawing, class filter.
- **Steps:** Load pretrained weights, run on webcam frames, filter by confidence, show FPS.
- **Show in the viva:** What confidence threshold and non-max suppression do.
- **Extensions:** Fine-tune on a few classes you label.

11. Lane Line Detection

- **Problem:** Find lane lines in road video.
- **Dataset:** Your own dashcam-style video.
- **Tech stack:** Python, OpenCV.
- **Main modules:** Region of interest, Canny edges, Hough lines, line averaging.
- **Steps:** Mask the road area, detect edges, fit Hough lines, average left and right lanes.
- **Show in the viva:** Why the region mask matters.
- **Extensions:** Handle curved lanes with polynomial fitting.

12. Panorama Stitcher

- **Problem:** Join overlapping photos into one panorama.
- **Dataset:** Your own overlapping photos.
- **Tech stack:** Python, OpenCV.
- **Main modules:** Feature detection (ORB or SIFT), matching, homography, blending.
- **Steps:** Find keypoints, match them, estimate homography with RANSAC, warp and blend.
- **Show in the viva:** What RANSAC removes.
- **Extensions:** Stitch more than three images.

Mini project report structure

A mini project report is usually 10 to 20 pages. Keep it short and clear:

1. **Title page and certificate** in your department's format.
2. **Introduction:** the problem in one paragraph and your objectives as a list.
3. **Tools used:** Python and OpenCV versions, camera used and lighting conditions.
4. **Design:** one block diagram or flowchart, plus the processing pipeline as a flowchart.
5. **Implementation:** what each module does, with 3 to 6 screenshots.
6. **Results:** screenshots of each stage and frames per second.
7. **Conclusion and future scope,** then references.

Ask your guide for the exact format before you start writing.

Common mistakes to avoid

- Hard-coding colour or threshold values that only work in one room.
- Showing only the final output and not the steps.
- Ignoring frames per second in a real-time demo.
- Using face images of others without asking.

Frequently asked questions

What is the easiest computer vision mini project?

A document scanner or a colour object tracker. Both use basic OpenCV functions and a webcam or phone photos.

Do I need deep learning for computer vision mini projects?

No. Many good mini projects use classic OpenCV methods. MediaPipe adds ready landmarks without training a model.

Which language is best for computer vision projects?

Python with OpenCV is the most common choice for students because of its libraries and examples.

Can I run these projects on a basic laptop?

Yes. Classic OpenCV and MediaPipe projects run in real time on most laptops without a GPU.

Does this PDF include full source code?

No. It gives the problem, data, stack, modules and steps for each project. You write the code yourself, which is what examiners test in the viva.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- OpenCV tutorials
- MediaPipe solutions guide
- Tesseract OCR on GitHub
- COCO dataset

About this PDF

Prepared by the Uncodemy training team as practice project ideas. Datasets and tools named are public; check each dataset's licence before use. We update it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550