PROJECT ABSTRACT

Fingerprint Spoof Detection using CNN

Project ID: k-1648 | projectech.in/projects/fingerprint-spoof-detection-cnn/

Fingerprint sensors can be fooled by a gelatin or silicone copy of a print — a presentation attack. This project trains a patch-based CNN on the LivDet competition dataset to distinguish live fingers from spoofs, evaluates it the hard way (tested on a spoof material never seen in training), and wraps it in a scanner demo with a liveness gauge and spoof-cue breakdown.

Suitable for B.E./B.Tech final-year projects in Computer Science, IT and AI & Machine Learning.

Project type: Software

Availability: Built to order

Suitable branches: Computer Science, IT and AI & Machine Learning

Technologies: Python 3, PyTorch, patch-based CNN, OpenCV, LivDet dataset, NumPy, scikit-learn, HTML/JS scanner demo

Price: Request a quotation

Prepared by Projectech · projectech.in — for academic project reference.

1. Title

Fingerprint Spoof Detection using CNN — a built-to-order software project delivered with a working demo application, complete source code, project report, presentation and viva preparation kit.

2. Introduction / Background

Liveness detection asks 'is this a real finger?', not 'whose finger?' — a distinct problem from matching. Spoofs betray themselves in texture: suppressed sweat pores, unnatural ridge uniformity. The strongest claim a student can make is cross-material generalization, and this project is built around proving it.

3. Problem Statement

Liveness detection asks “is this a real finger?”, not “whose finger?” — a distinct problem from matching. Spoofs betray themselves in texture: suppressed sweat pores, unnatural ridge uniformity. The strongest claim a student can make is cross-material generalization, and this project is built around proving it.

4. Objectives

- Assemble LivDet live and spoof captures per the competition protocol.
- Build a patch-based CNN liveness classifier with majority-vote fusion.
- Evaluate cross-material: train on three spoof materials, test on the held-out fourth.
- Reach the design targets: APCER ≤ 4.5% at BPCER = 1%, accuracy ≥ 96%.

- Deliver a scanner demo with liveness gauge, cue breakdown and event logging.

5. Proposed Methodology

LivDet captures (gelatin, silicone, latex, wood-glue spoofs across sensors) are enhanced with Gabor filtering, segmented and split into overlapping patches. A CNN binary classifier trains on patches with rotation and elastic-distortion augmentation, using splits that hold out one spoof material entirely. At inference, patch scores fuse by majority vote into a 0–100 liveness score; parallel cue analyzers (pore visibility, ridge continuity, texture variance) explain each decision.

6. System Architecture / Working

A Python pipeline (preprocessing, training, cross-material evaluation, DET curves) feeds a single-file HTML/JS scanner demo: the scan tab animates capture with a liveness gauge and grant/deny verdict; the cues tab breaks down per-cue measurements; the evaluation tab shows the DET curve and per-material APCER.

7. Hardware / Software Requirements

- Hardware: any machine that runs Python 3 (4 GB RAM minimum; CPU inference suffices).
- Software: Python 3, PyTorch, OpenCV, NumPy, Matplotlib, scikit-learn.
- Data: LivDet competition dataset (free for research).
- Scanner demo: any modern web browser; runs offline after download.

8. Expected Outcomes

- A liveness classifier meeting the design APCER/BPCER targets, including on unseen materials.
- A scanner demo that explains each decision through measurable spoof cues.
- Cross-material evaluation proving generalization beyond memorized materials.
- A documented presentation-attack threat model for the viva.

9. Applications

- Academic final-year project in Computer Science, IT or AI & Machine Learning programs.
- Teaching aid for biometric security and patch-based CNN design.
- Prototype liveness module for an access-control or attendance product.

10. Future Scope

- Fusion with a fingerprint matcher into a complete authentication stack.
- Additional modalities (face, iris presentation-attack detection).
- On-sensor deployment with a hardware SDK.

- Unknown-material detection via anomaly scoring.