Machine Vision Lab PM: Galactic Credit Scanner Mission

Background Story

The year is 35 ABY, and the galaxy is in turmoil. The remnants of the First Order have collapsed, but chaos reigns in the Outer Rim. Smugglers, bounty hunters, and traders flood the hyperspace lanes, exchanging goods for Galactic Credits—the universal currency of the Star Wars galaxy.

However, counterfeit credits have begun to circulate, threatening the fragile economy of the New Republic. The Galactic Banking Clan has hired you, an elite Machine Vision Engineer, to develop an automated system that can scan, identify, and track authentic credit chips as they move along high-speed conveyors in interstellar trade hubs.

Your mission:
- Build a vision system that can detect credit chips by color, if they are fake or real, determine their denomination, and calculate total value in real time.
- The fate of galactic commerce depends on your accuracy—failure means financial collapse across star systems.

May the Force of OpenCV be with you.

Objective

Develop a Python + OpenCV program that uses a camera and conveyor belt to detect, identify, and track 3D-printed Star Wars credits based on color and pattern to calculate their value.

Equipment

- Conveyor belt system
- Camera (Webcam or Basler)
- Computer with Python and OpenCV installed
- 3D-printed Star Wars credits in three colors (each color = different value)

Instructions

1. Setup:

   - Mount the camera above the conveyor belt.
   - Ensure proper lighting to minimize shadows and reflections.
   - - Calibrate the camera for optimal resolution and frame rate.

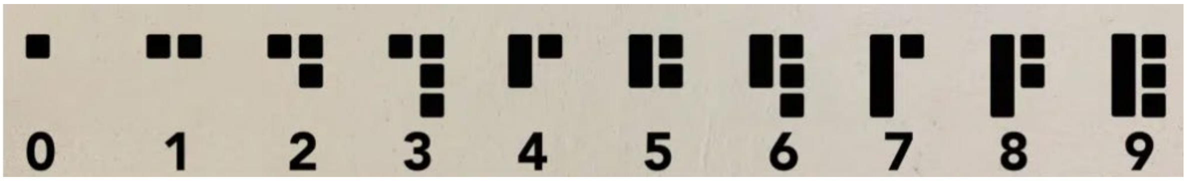
2. Programming Requirements:

   - Use Python and OpenCV for image acquisition and processing.
   - Implement color-based detection to classify credits.
   - Handle varying positions and orientations of credits on the conveyor.
   - AI is allowed to be used

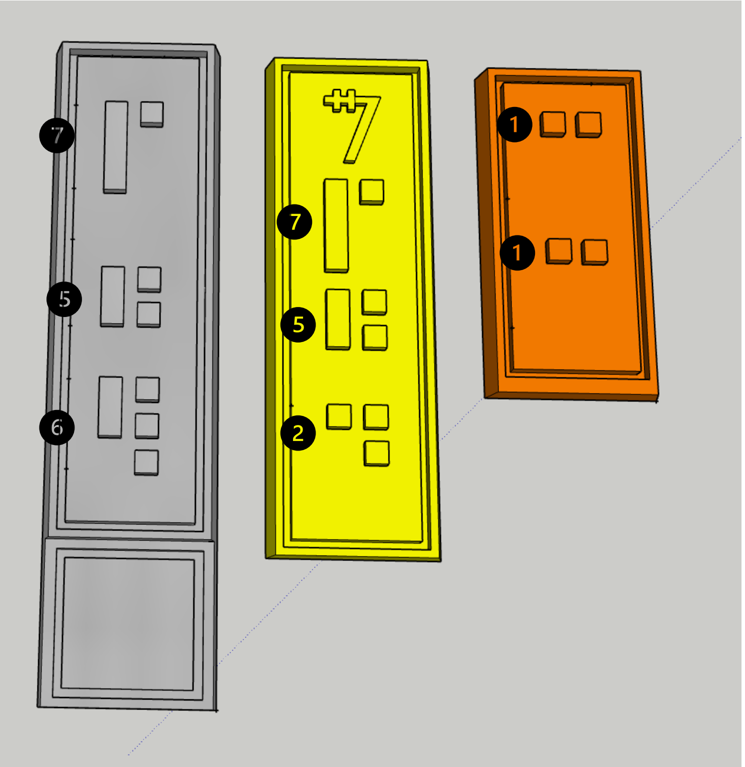
3. Notes for Students

   - Define **color-to-value mapping** clearly (e.g., Red = 10 credits, Blue = 5 credits, Green = 1 credit).
   - Ensure **consistent lighting**; avoid glare and shadows for accurate color detection.
   - Mount the **camera directly above the conveyor**; consider frame rate for real-time tracking.
   - **Performance expectations**: Aim for at least 10–15 FPS for real-time grades (10-20%).
   - Plan **error handling** for partially visible or overlapping chips.
   - Start testing with **static images** before moving to video; create a small dataset for validation.
   - Display results clearly on the video feed (e.g., overlay text with credit values and total sum).
   - Take pictures and make a recording of the feed you can work with at home.

Credit Value system:



How to decode each credit:

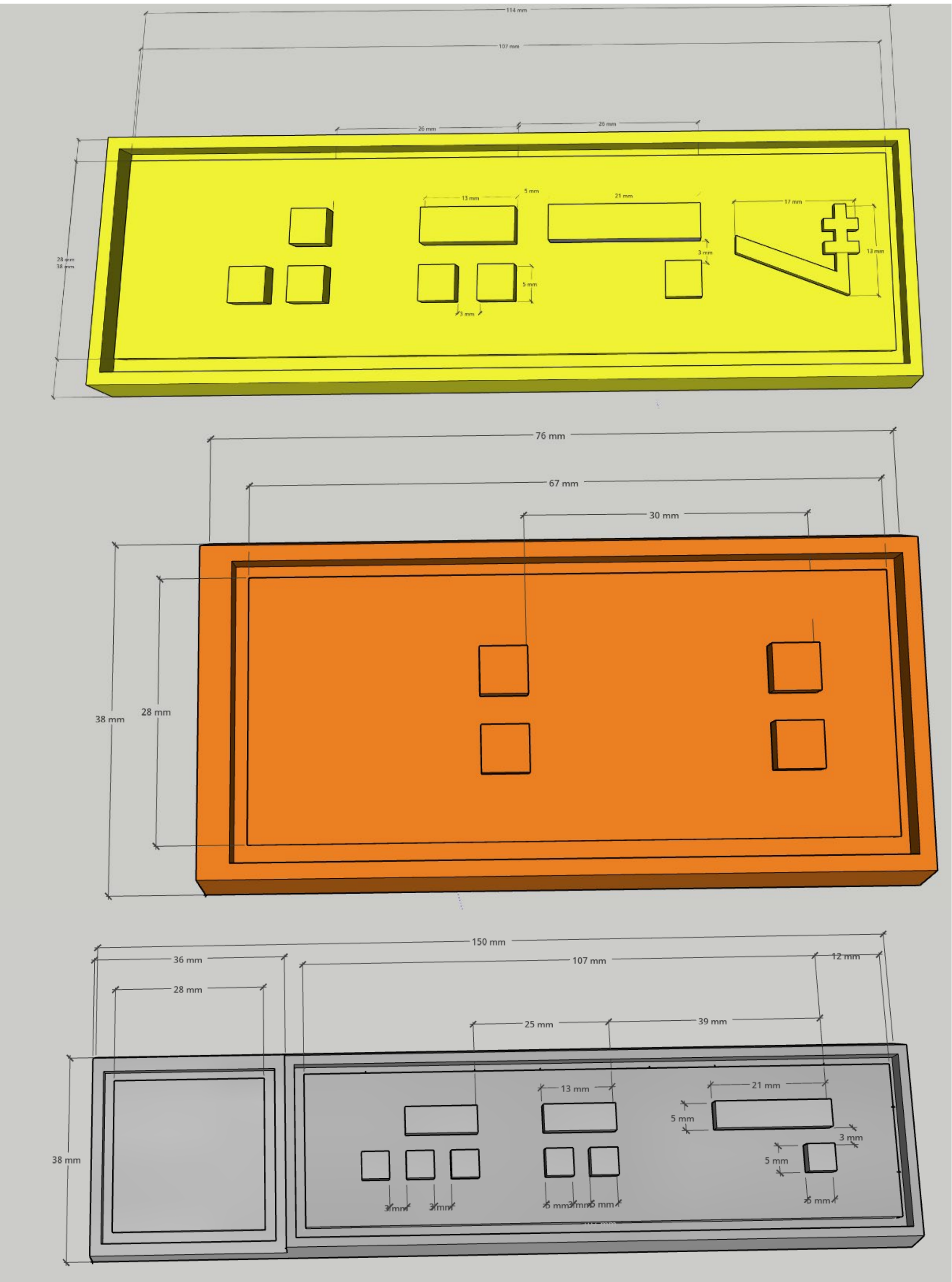


Bronze: Take each value and multiply them, in this case 1*1=1.

Silver: Read from top down, the value is the number-series in this case 756.

Gold: Read from top down, the value is the number series multiplied by 10, in this case 7520.

Credit measurements:



Grading Criteria

Grade & Rank	Requirement
F – Bantha Fodder (Fail)	No functional detection.
E – Padawan (pass)	Detect a single credit and correctly identify its value an image feed.
C – Jedi Knight (10 % bonus on the exam)	Detect multiple credits in a static image and calculate total value and if it a fake.
A – Grand Master of the Jedi Order (20 % bonus on the exam)	Real-time detection, fake detection, tracking, and dynamic summation of all credits on screen during conveyor operation.

Deliverables

- Live demonstration of the program according to the targeted grade.
- Upload source code after demonstration on canvas.

Yodas Methodical Tips

1. **Break the task, you must.**
    - Stage 1: Image Acquisition, first, capture the image. Strong foundation, this is.
    - Stage 2: Preprocessing, prepare the image next. Noise, banish you will.
    - Stage 3: Segmentation, segment by color. In HSV or BGR, trust you should.
    - Stage 4: Feature Extraction, features, extract. Contours and positions, find.
    - Stage 5: Classification , classify colors. Credits, together with certain features, their value reveal.
    - Stage 6: Tracking, track them, across frames. Identity, maintain you must.
    - Stage 7: Aggregation & Display, Sum and display. Truth of the credits, show you shall.
2. **Test each step, you will.**
    - Static images first, then video.
    - Rush not. Patience, a Jedi needs.
3. **Simple, begin. Complex, later.**
    - One color, one chip.
    - Many colors, many chips.
    - Real-time mastery, at the end.
4. **Robustness, plan for it you must.**
    - Lighting changes, conveyor speed.
    - Modular code, your ally it is.
5. **Assumptions, write down.**
    - Color ranges, camera settings, limitations.
    - Clear your mind, clear your notes.