

1) Libraries used

cv2 (OpenCV)

- used to access webcam, read video frames, change resolution, display image windows.
- OpenCV provides low level computer vision tools.

cvzone

- cvzone is built on top of OpenCV + Mediapipe.
- makes human-computer-vision tasks easier with high level ready modules.
- HandDetector class detects human hands in real time and returns landmarks, bounding boxes etc.

math

- used for simple hand distance measurement between 2 points.

2) How Hand tracking works conceptually

cvzone internally uses **Mediapipe Hand Landmark Model** (21 key points on hand).

Each frame → cvzone:

- finds hand in image
- returns results like
 - lmList → list of 21 hand landmark exact pixel coordinates (x,y,z)
 - bbox → bounding box around the hand (x, y, width, height)
 - center → center of bounding box / approximate center of palm

Then we use these to detect gestures.

3) Technique used here

Gesture based VFX overlay

The Dr. Strange shield appears **only when index finger is up.**

`detector.fingersUp(hand)` returns a list like [thumb, index, middle, ring, pinky] → 1 if finger up, 0 if finger down.

You check if index finger is up:

```
if 1 in detector.fingersUp(hand):
```

& shield is drawn on that finger and moves / scales according to hand position.

4) video overlay technique

We load another video: "effect.mp4" — the shield VFX.

We read frames from that video continuously and overlay them on the webcam.

To place overlay correctly:

- we detect hand center (cx, cy)
- detect hand size via bbox width ($handSize$)
- scale effect based on $handSize \rightarrow$ closer hand \rightarrow bigger shield.

overlay math:

```
final_image = (1 - mask)*background + overlay
```

This blends the shield effect naturally.

5) toggling shield on/off using two hand gesture

This part is cool and important.

- If **both hands are detected**
- and **landmark 9 on both hands** are close (distance < 30 pixels)
- and user holds it long enough (changeTimer > 100)

→ then shield visibility toggles.

This means user gesture controls VFX mode.

6) Why unpacking was important

```
hands, img = detector.findHands(img, draw=False)
```

cvzone returns 2 things:

- modified image (with drawing)
- list of hand results

We needed to store both separate.

7) What computer vision concepts used here

Concept	Where used
Webcam Capture	<code>cap = cv2.VideoCapture(0)</code>
Frame processing real-time loop	<code>while True</code>
Hand landmark model	<code>HandDetector()</code>
Gesture recognition	<code>fingersUp(hand)</code>
Multi-hand detection	<code>if len(hands) == 2</code>
Euclidean Distance	<code>findDistance(...)</code>
Alpha blending overlay	<code>(1-mask)*bg + overlay</code>
Dynamic scaling based on distance	<code>mapFromTo()</code>

This entire script is basically **gesture controlled VFX compositing**.

This is exactly how AR / XR / Interactive filters (Snapchat lens / TikTok filters) work internally.

Code Summary

This program uses a webcam + hand tracking to create a Doctor Strange magical shield effect in real-time.

- It captures video from the webcam using OpenCV.
- It loads a VFX video (effect.mp4) which will be used as the magic shield overlay.
- It uses **cvzone HandDetector** (built on Mediapipe) to detect hands and identify hand landmarks + finger positions.
- If the index finger is raised → the effect (shield) is drawn at the hand's position.
- The size of the effect scales automatically based on the detected hand size.
- If two hands come close together (distance between specific points is small) → the shield toggles ON/OFF after a short delay.
- Alpha blending is used to overlay the shield transparently on the video frame.
- Final output is displayed live in a window titled “Doctor Strange”.

In short

Hand gestures control whether a magical shield visual effect appears and where it is placed on the screen, based on the hand's position and finger state.