```
#  Augmented Reality Marker Detection – image input images
aruco29,aruco35,aruco49.png


Import cv2

Import cv2.aruco as aruco


Def detect_aruco_markers(image):

   Dictionaries = [

      Aruco.DICT_4X4_50,

      Aruco.DICT_4X4_100,

      Aruco.DICT_4X4_250,

      Aruco.DICT_4X4_1000,

      Aruco.DICT_5X5_50,

      Aruco.DICT_5X5_100,

      Aruco.DICT_5X5_250,

      Aruco.DICT_5X5_1000,

      Aruco.DICT_6X6_50,

      Aruco.DICT_6X6_100,

      Aruco.DICT_6X6_250,

      Aruco.DICT_6X6_1000,

      Aruco.DICT_7X7_50,

      Aruco.DICT_7X7_100,

      Aruco.DICT_7X7_250,

      Aruco.DICT_7X7_1000,

      Aruco.DICT_ARUCO_ORIGINAL

   ]
```

```
    Gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    For dict_id in dictionaries:
        Aruco_dict = aruco.getPredefinedDictionary(dict_id)
        Parameters = aruco.DetectorParameters()

        Corners, ids, _ = aruco.detectMarkers(gray, aruco_dict, parameters=parameters)

        If ids is not None:
            Print(f"Detected with dictionary: {dict_id}")
            Aruco.drawDetectedMarkers(image, corners, ids)
            For I in range(len(ids)):
                Corner = corners[i][0]
                Top_left_x = int(corner[0][0])
                Top_left_y = int(corner[0][1])
                Cv2.putText(image, f"ID: {ids[i][0]}", (top_left_x, top_left_y – 10),
                        Cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
                Print(f"Detected marker ID: {ids[i][0]}")
            Return image

    Print("No marker detected.")
    Return image


# Usage example:
```

```
Image_path = “aruco49.png”  # Change to your image path
Frame = cv2.imread(image_path)

If frame is None:
   Print(“Could not read the image. Please check the path.”)
   Exit()

# Optionally resize for display
Frame = cv2.resize(frame, (500, 500))

# Detect markers from multiple dictionaries
Output_image = detect_aruco_markers(frame)

# Show the result
Cv2.imshow(“Detected ArUco Markers”, output_image)
Cv2.waitKey(0)
Cv2.destroyAllWindows()
```

```
#  Augmented Reality Marker Detection – webcam input

Import cv2
Import cv2.aruco as aruco
```

```
Def detect_aruco_markers(image):
  Dictionaries = [
    Aruco.DICT_4X4_50,
    Aruco.DICT_4X4_100,
    Aruco.DICT_4X4_250,
    Aruco.DICT_4X4_1000,
    Aruco.DICT_5X5_50,
    Aruco.DICT_5X5_100,
    Aruco.DICT_5X5_250,
    Aruco.DICT_5X5_1000,
    Aruco.DICT_6X6_50,
    Aruco.DICT_6X6_100,
    Aruco.DICT_6X6_250,
    Aruco.DICT_6X6_1000,
    Aruco.DICT_7X7_50,
    Aruco.DICT_7X7_100,
    Aruco.DICT_7X7_250,
    Aruco.DICT_7X7_1000,
    Aruco.DICT_ARUCO_ORIGINAL
  ]

  Gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

  For dict_id in dictionaries:
    Aruco_dict = aruco.getPredefinedDictionary(dict_id)
```

```
    Parameters = aruco.DetectorParameters()

    Corners, ids, _ = aruco.detectMarkers(gray, aruco_dict, parameters=parameters)

    If ids is not None:
      Print(f”Detected with dictionary: {dict_id}”)
      Aruco.drawDetectedMarkers(image, corners, ids)
      For I in range(len(ids)):
        Corner = corners[i][0]
        Top_left_x = int(corner[0][0])
        Top_left_y = int(corner[0][1])
        Cv2.putText(image, f”ID: {ids[i][0]}”, (top_left_x, top_left_y – 10),
              Cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
        Print(f”Detected marker ID: {ids[i][0]}”)
      Return image

  Return image


Def main():
  Cap = cv2.VideoCapture(0)

  If not cap.isOpened():
    Print(“Cannot open webcam”)
    Exit()
```

```
    While True:
        Ret, frame = cap.read()
        If not ret:
            Print(“Failed to grab frame”)
            Break

        # Resize frame if needed (optional)
        Frame = cv2.resize(frame, (640, 480))

        # Detect ArUco markers
        Output_frame = detect_aruco_markers(frame)

        # Show the frame with detected markers (if any)
        Cv2.imshow(“Webcam ArUco Detection”, output_frame)

        # Press ‘q’ to quit
        If cv2.waitKey(1) & 0xFF == ord(‘q’):
            Break

    Cap.release()
    Cv2.destroyAllWindows()

If __name__ == “__main__”:
    Main()
```