

# **PAPER VISION AUGMENTED REALITY**

**Project Based Learning -IV (22AI012)**

*Submitted by*

**SONALI SAINI**

**(2210993848)**

**SHAIL SHARMA**

**(2210993837)**

**Semester: 4**



**BE-CSE (Artificial Intelligence)**

*Guided by*

**Dr. Cheshta Dhingra**

**CHITKARA UNIVERSITY INSITUTE OF ENGINEERING & TECHNOLOGY**

**CHITKARA UNIVERSITY, RAJPURA**

**MAY,2024**

# Acknowledgement

With immense pleasure we, Ms. Sonali Saini (2210993848) and Mr. Shail Sharma (2201993837) presenting “Paper Vision Augmented Reality” project report as part of curriculum of ‘BE-CSE(AI)’

We would like to express our sincere thanks to Dr. Cheshta Dhingra, for her valuable guidance and support in completing my project.

We would also like to express my gratitude towards our dean Dr. Sushil Narang for giving us this great opportunity to do a project on Paper Vision Augmented Reality.

Without their support and suggestions, this project would not have been completed.

Signature:

Name: Sonali Saini

Roll No: 2210993848

Signature:

Name: Shail Sharma

Roll No: 2210993837

# **Table of Contents**

1. Abstract
2. Introduction
3. Problem Formulation
4. Methodology
5. Flowchart
6. Software and Hardware Requirements
7. Code
8. Output/Results
9. Conclusion and Future scope
10. References

# 1. Abstract

This project explores the fusion of Paper Vision augmented reality (AR) technology with advanced machine learning (ML) methodologies to create an immersive and dynamic AR application. The core objective is to develop ML models capable of marker less object recognition, real-time tracking, and seamless integration of dynamic content within the AR environment. The methodology involves comprehensive data collection, meticulous annotation, and preprocessing techniques essential for training robust ML algorithms. The selected ML models undergo rigorous training and optimization to ensure high accuracy, real-time performance, and user-centric interactions in the AR application.

The AR environment is meticulously crafted using Paper Vision, incorporating sophisticated marker less tracking mechanisms and seamless integration of ML models for intelligent content placement. Performance optimization strategies are implemented to achieve optimal ML inference speed, accuracy, and responsiveness, ensuring a compelling user experience. The project's findings demonstrate the potential of ML-driven AR experiences across various domains, highlighting opportunities for innovative applications in education, training simulations, and interactive retail experiences.

Looking ahead, the project's future scope includes enhancing ML models for handling complex interactions, optimizing scalability and deployment for diverse platforms, and fostering collaboration for continued innovation in ML-enhanced AR technologies. By pushing the boundaries of immersive digital experiences, this project contributes to the evolution of AR applications, paving the way for transformative advancements in interactive and intelligent AR environments.

## 2. Introduction

Augmented Reality (AR) has significantly evolved in recent years, transforming how we interact with digital content by seamlessly blending virtual elements into our physical environment. This technology has widespread applications across industries, from entertainment and gaming to healthcare and education. However, conventional AR systems often rely on predefined markers or patterns for object recognition, limiting their adaptability and real-time responsiveness in dynamic environments.

On the other hand, Machine Learning (ML) algorithms offer a revolutionary approach to processing information, enabling computers to learn from data, recognize patterns, and make predictions. The integration of Paper Vision, a leading AR framework, with advanced ML techniques presents a compelling opportunity to enhance AR experiences significantly. By leveraging ML models for tasks such as object detection, classification, and tracking within the Paper Vision environment, we can create more intelligent and context-aware AR applications.

The primary goal of this project is to showcase the synergies between Paper Vision augmented reality and Machine Learning algorithms, highlighting their combined potential for creating immersive and interactive AR environments. Our objectives include implementing ML models for real-time object recognition and tracking, enhancing user interactions through intelligent content placement, evaluating performance metrics, and exploring diverse applications across industries.

To achieve these goals, our methodology encompasses a systematic approach involving data collection and preprocessing for ML model training, selection of appropriate ML algorithms, development of the AR application using Paper Vision, integration of ML models for object recognition, and rigorous testing and validation to refine the system's performance.

The expected outcomes of this project include a functional AR application demonstrating ML-powered object recognition and tracking capabilities, performance metrics indicating the accuracy and usability of ML-enhanced AR features, and insights into potential use cases and industries where ML-driven AR solutions can create value. This project lays the foundation for future research in ML-driven AR technologies, paving the way for advancements in spatial computing and interactive digital experiences.

### 3. Problem formulation

**2.1 Marker less Object Recognition:** Traditional AR systems often rely on markers or predefined patterns for object recognition. However, this approach can be limiting in dynamic environments where objects may not have predefined markers. The project seeks to develop ML algorithms capable of marker less object recognition and tracking, allowing for more flexible and adaptive AR experiences.

**2.2 Real-time Performance:** Achieving real-time performance is critical for AR applications to deliver seamless and immersive user experiences. Integrating ML models into the AR pipeline introduces computational overhead, which must be optimized to ensure smooth rendering and interaction. The project aims to tackle this challenge by optimizing ML inference processes and minimizing latency.

**2.3 Context-aware Interactions:** Enhancing user interactions in AR environments involves understanding the context and intent behind user actions. ML techniques can be leveraged to create context-aware AR applications that intelligently place virtual content, respond to user gestures, and personalize experiences based on user preferences and environmental cues.

**2.4 Scalability and Generalization:** Developing ML-driven AR solutions that are scalable and generalize well across different scenarios and environments is another challenge. The project seeks to explore methods for training ML models on diverse datasets to improve robustness, adaptability, and generalization of AR applications within the Paper Vision framework.

By addressing these challenges, the project aims to advance the state-of-the-art in ML enhanced AR systems, unlocking new possibilities for interactive, intelligent, and context aware virtual experiences across various domains.

## **4. Methodology**

### **1. Data Preparation:**

- 1.1 Collect diverse datasets representing real-world scenarios.
- 1.2 Annotate data for object recognition accuracy.
- 1.3 Preprocess data for ML training, including cleaning and augmentation.

### **2. Model Selection and Training:**

- 2.1 Choose ML algorithms suitable for the project's goals.
- 2.2 Train models using annotated data sets.
- 2.3 Optimize models for accuracy, robustness, and efficiency.

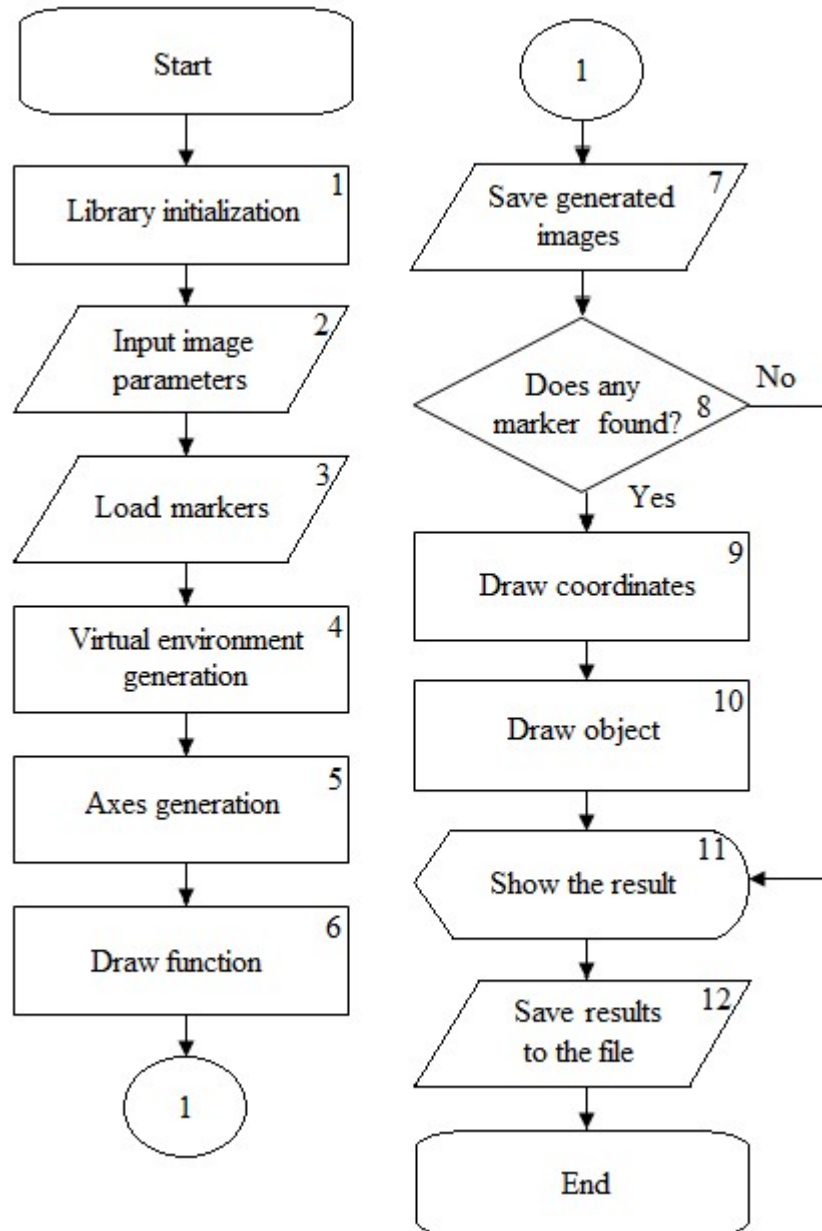
### **3. AR Environment Setup:**

- 3.1 Utilize Paper Vision or similar AR platforms.
- 3.2 Implement marker less tracking for real-time object positioning.
- 3.3 Integrate ML models for dynamic content in the AR environment.

### **4. Performance Optimization:**

- 4.1 Optimize ML inference for real-time responsiveness.
- 4.2 Evaluate model accuracy and gather user feedback.
- 4.3 Refine ML models and AR interactions based on performance metrics.

## 5. Flowchart





## 6. Software and Hardware Requirements

### Software Requirements:

1. Image and video editing software (e.g., Adobe Photoshop, GIMP)
2. Python libraries for preprocessing (OpenCV, NumPy)
3. Machine Learning frameworks (TensorFlow, PyTorch)
4. Integrated Development Environments (IDEs) (VS Code)
5. Libraries for model optimization (scikit-learn, TensorFlow Hub)
6. AR development platforms (Paper Vision, Unity 3D, Unreal Engine)
7. AR Software Development Kits (SDKs)
8. Profiling tools (Tensor Board) for performance analysis
9. Version control systems (Git, GitHub)

### Hardware Requirements:

1. Camera
2. Storage devices for datasets
3. High-performance CPUs or GPUs
4. Cloud computing resources (Google Cloud Platform, AWS EC2 instances)
5. AR-compatible devices (smartphones, tablets)
6. VR headsets for immersive testing (Oculus Rift)
7. Mobile devices (Android) for testing ML inference
8. Development workstations or laptops with sufficient processing power and memory
9. Testing devices for AR application deployment and user feedback gathering

## 7. Code: -

```
main_1.py

import cv2
import numpy as np
import time
import os
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('--video', help='Path to the video to be projected',
                    default='Ronaldo.mp4')

args = parser.parse_args()
video_path=args.video
videostream = cv2.VideoCapture(0)
video = cv2.VideoCapture(video_path)
dictionary = cv2.aruco.Dictionary_get(cv2.aruco.DICT_6X6_250)
parameters = cv2.aruco.DetectorParameters_create()

def find_and_warp(frame, source, cornerIds, dictionary, parameters):
    (imgH, imgW) = frame.shape[:2]
    (srcH, srcW) = source.shape[:2]
    markerCorners, markerIds, rejectedCandidates = cv2.aruco.detectMarkers(frame, dictionary,
    parameters=parameters)
    if len(markerCorners)!= 4:
        markerIds = np.array([])
    else:
        markerIds.flatten()
    refPts = []
    for i in cornerIds:
        j = np.squeeze(np.where(markerIds == i))
        if j.size == 0:
            continue
        else:
            j = j[0]
        markerCorners = np.array(markerCorners)
        #print(markerCorners)
        corner = np.squeeze(markerCorners[j])
        refPts.append(corner)
    if len(refPts) != 4:
        return None
    (refPtTL, refPtTR, refPtBR, refPtBL) = np.array(refPts)
    dstMat = [refPtTL[0], refPtTR[1], refPtBR[2], refPtBL[3]]
    dstMat = np.array(dstMat)
    srcMat = np.array([[0, 0], [srcW, 0], [srcW, srcH], [0, srcH]])
    (H, _) = cv2.findHomography(srcMat, dstMat)
    warped = cv2.warpPerspective(source, H, (imgW, imgH))
    mask = np.zeros((imgH, imgW), dtype="uint8")
    cv2.fillConvexPoly(mask, dstMat.astype("int32"), (255, 255, 255),
    cv2.LINE_AA)
    maskScaled = mask.copy() / 255.0
    maskScaled = np.dstack([maskScaled] * 3)
    warpedMultiplied = cv2.multiply(warped.astype("float"),
    maskScaled)
    imageMultiplied = cv2.multiply(frame.astype(float),
    1.0 - maskScaled)
    output = cv2.add(warpedMultiplied, imageMultiplied)
    output = output.astype("uint8")
    return output

while True:
    ret, frame = videostream.read()
    ret, source = video.read()
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR)
    frame_expanded = np.expand_dims(frame_rgb, axis=0)
    scale_percent = (1000/frame.shape[0])*100
    width = int(frame.shape[1] * scale_percent / 100)
    height = int(frame.shape[0] * scale_percent / 100)
    dim = (width, height)
    frame = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)
    width2 = int(source.shape[1] * scale_percent / 100)
    height2 = int(source.shape[0] * scale_percent / 100)
    dim = (width2, height2)
    source = cv2.resize(source, dim, interpolation = cv2.INTER_AREA)
    warped = find_and_warp(
        frame, source,
        cornerIds=(1, 2, 4, 3),
        dictionary=dictionary,
        parameters=parameters,
    )
    if warped is not None:
        frame = warped
    cv2.imshow('Augmented Reality', frame)
    if cv2.waitKey(1) == ord('q'):
        break

cv2.destroyAllWindows()
print("Done")
```

main\_2.py

```
from flask import Flask, url_for, render_template, request, redirect, session
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///users.db'
db = SQLAlchemy(app)

class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(100), unique=True)
    password = db.Column(db.String(100))
    def __init__(self, username, password):
        self.username = username
        self.password = password

@app.route('/Log-in/index.html', methods=['GET'])
def index_login():
    if session.get('logged_in'):
        return render_template('home.html')
    else:
        return render_template('index.html', message="LOGIN To AR Video Visualizer")
@app.route('/register/', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        try:
            db.session.add(User(username=request.form['username'],
password=request.form['password']))
            db.session.commit()
            return redirect(url_for('login'))
        except:
            return render_template('index.html', message="User Already Exists")
    else:
        return render_template('register.html')
@app.route('/login/', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        return render_template('login.html')
    else:
        u = request.form['username']
        p = request.form['password']
        data = User.query.filter_by(username=u, password=p).first()
        if data is not None:
            session['logged_in'] = True
            return redirect(url_for('index'))
        return render_template('index.html', message="Incorrect Details")

@app.route('/logout', methods=['GET', 'POST'])
def logout():
    session['logged_in'] = False
    return redirect(url_for('index'))
@app.route('/')
def index():
    return render_template('index_main.html')
@app.route('/about.html')
def about_html():
    return render_template('about_main.html')
@app.route('/service.html')
def service():
    return render_template('service_main.html')
@app.route('/why_main')
def why_html():
    return render_template('why_main.html')
if __name__ == '__main__':
    app.secret_key = "ThisIsNotASecret:p"
    with app.app_context():
        db.create_all()
    app.run(debug=True)
```

```

main_3.py

<!DOCTYPE html>
<html>
<body class="sub_page"

    <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav">
            <li class="nav-item">
                <a class="nav-link" href="index.html">Home </a>
            </li>
            <li class="nav-item active">
                <a class="nav-link" href="about.html"> About <span class="sr-only">(current)
            </span> </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="service.html">Services</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="why.html">Why Us</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="">Model</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="../Log-in/index.html"> <i class="fa fa-user" aria-
hidden="true"></i> Login</a>
            </li>
            <form class="form-inline">
                <!-- <button class="btn my-2 my-sm-0 nav_search-btn" type="submit"> -->
                <!-- <i class="fa fa-search" aria-hidden="true"></i> -->
            </button>
            </form>
        </ul>
    </div>
</nav>
</div>
</header>
<!-- end header section -->
</div>

<!-- about section -->

<section class="about_section layout_padding">
    <div class="container">
        <div class="heading_container heading_center">
            <h2>
                About <span>Us</span>
            </h2>
            <p>
                We are innovators merging AR and OpenCV to redefine interactive visual experiences
            </p>
        </div>
        <div class="row">
            <div class="col-md-6">
                <div class="img-box">
                    
                </div>
            </div>
            <div class="col-md-6">
                <div class="detail-box">
                    <h3>
                        We Are AR Video Visualizer
                    </h3>
                    <p>
                        At AR Video Visualizer, we're passionate about pushing the boundaries of
augmented reality technology.
                        Leveraging the power of OpenCV-Python, we aim to create immersive experiences
where digital content seamlessly integrates with the real world.
                        Our team is dedicated to developing innovative solutions for dynamic QR code
generation, real-time video rendering, and precise object recognition.
                    </p>
                    <p>
                        Join us on our journey to redefine visual interaction and explore new frontiers
in augmented reality
                    </p>
                    <!-- <a href=""> -->
                    <!-- Read More -->
                    </a>
                </div>
            </div>
        </div>
    </div>
</section>

<!-- end about section -->

<!-- info section -->

<section class="info_section layout_padding2">
    <div>
    </div>
</div>
</section>

<!-- end info section -->

<!-- footer section -->
<section class="footer_section">
    <div class="container">
        <p>
            &copy; <span id="displayYear"></span> All Rights Reserved By
            <a href="https://html.design/">AR Video Visualizer
        </a>
        </p>
    </div>
</section>
<!-- footer section -->
</body>
</html>

```



main\_3.py

```
import cv2
import numpy as np

dictionary = cv2.aruco.Dictionary_get(cv2.aruco.DICT_6X6_250)

markerImage = np.zeros((200, 200), dtype=np.uint8)
markerImage = cv2.aruco.drawMarker(dictionary, 1, 200, markerImage, 1)

cv2.imwrite("arucomarker.png", markerImage)
```







```

main.css

/*
 * Bootstrap v4.3.1 (https://getbootstrap.com/)
 * Copyright 2011-2019 The Bootstrap Authors
 * (https://github.com/twbs/bootstrap/graphs/contributors)
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 */

(function (global, factory) {
  typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports,
    require('jquery'), require('popper.js')) :
    typeof define === 'function' && define.amd ? define(['exports', 'jquery', 'popper.js'],
      factory) :
      (global = global || self, factory(global.bootstrap = {}, global.jQuery, global.Popper));
})(this, function (exports, $, Popper) {
  'use strict';

  $ = $ && $.hasOwnProperty('default') ? $['default'] : $;
  Popper = Popper && Popper.hasOwnProperty('default') ? Popper['default'] : Popper;

  function _defineProperties(target, props) {
    for (var i = 0; i < props.length; i++) {
      var descriptor = props[i];
      descriptor.enumerable = descriptor.enumerable || false;
      descriptor.configurable = true;
      if ("value" in descriptor) descriptor.writable = true;
      Object.defineProperty(target, descriptor.key, descriptor);
    }
  }

  function _createClass(Constructor, protoProps, staticProps) {
    if (protoProps) _defineProperties(Constructor.prototype, protoProps);
    if (staticProps) _defineProperties(Constructor, staticProps);
    return Constructor;
  }

  function _defineProperty(obj, key, value) {
    if (key in obj) {
      Object.defineProperty(obj, key, {
        value: value,
        enumerable: true,
        configurable: true,
        writable: true
      });
    } else {
      obj[key] = value;
    }
  }

  return obj;
}

function _objectSpread(target) {
  for (var i = 1; i < arguments.length; i++) {
    var source = arguments[i] != null ? arguments[i] : {};
    var ownKeys = Object.keys(source);

    if (typeof Object.getOwnPropertySymbols === 'function') {
      ownKeys = ownKeys.concat(Object.getOwnPropertySymbols(source).filter(function (sym) {
        return Object.getOwnPropertyDescriptor(source, sym).enumerable;
      }));
    }

    ownKeys.forEach(function (key) {
      _defineProperty(target, key, source[key]);
    });
  }

  return target;
}

function _inheritsLoose(subClass, superClass) {
  subClass.prototype = Object.create(superClass.prototype);
  subClass.prototype.constructor = subClass;
  subClass.__proto__ = superClass;
}

/**
 * -----
 * Bootstrap (v4.3.1): util.js
 * Licensed under MIT (https://github.com/twbs/bootstrap/blob/master/LICENSE)
 * -----
 */

var TRANSITION_END = 'transitionend';
var MAX_UID = 1000000;
var MILLISECONDS_MULTIPLIER = 1000; // Shoutout AngusCroll (https://goo.gl/pxwQGp)

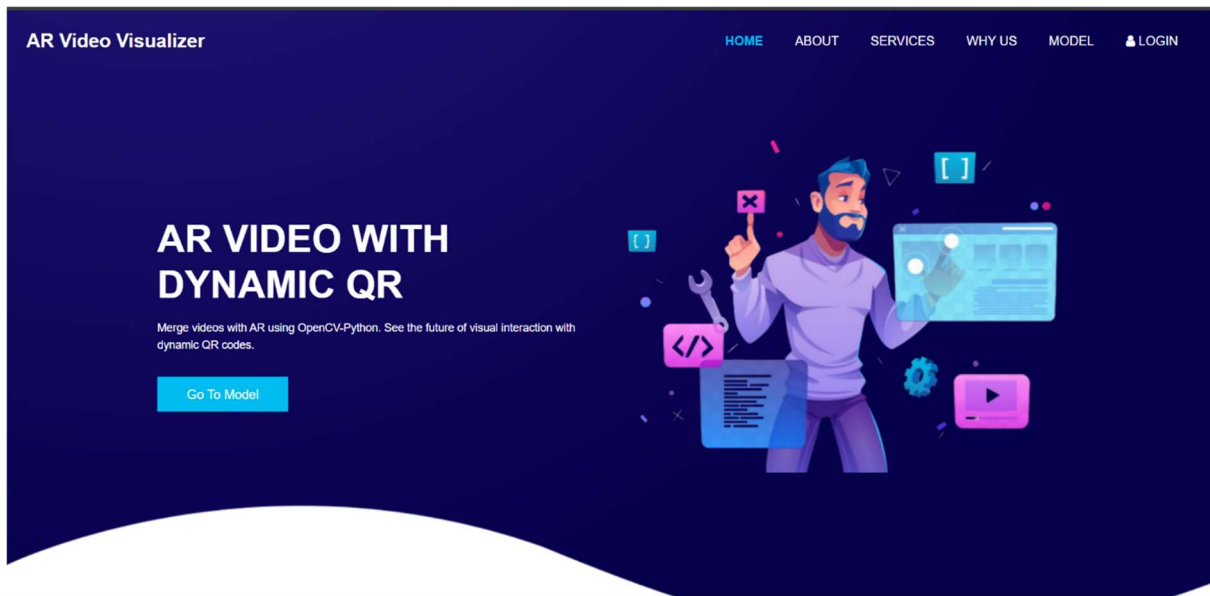
function toType(obj) {
  return {}.toString.call(obj).match(/\s([a-z]+)/i)[1].toLowerCase();
}

function getSpecialTransitionEndEvent() {
  return {
    bindType: TRANSITION_END,
    delegateType: TRANSITION_END,
    handle: function handle(event) {
      if ($(event.target).is(this)) {
        return event.handleObj.handler.apply(this, arguments); // eslint-disable-line
        prefer-rest-params
      }
    }
  };
}

return {
  TRANSITION_END: TRANSITION_END,
  MAX_UID: MAX_UID,
  MILLISECONDS_MULTIPLIER: MILLISECONDS_MULTIPLIER,
  toType: toType,
  getSpecialTransitionEndEvent: getSpecialTransitionEndEvent
};

```

## 8. Outputs/Results



### Our Services

Real-time AR video rendering, dynamic QR code generation, and precise OpenCV-powered recognition



#### REAL-TIME VIDEO RENDERING

Display videos in augmented reality, seamlessly integrating digital content into the real world



#### DYNAMIC QR CODE GENERATION

Generate and display QR codes dynamically within the AR environment, enabling interactive experiences



#### OPENCV-POWERED RECOGNITION

Utilize OpenCV-Python for accurate and efficient recognition of QR codes, ensuring seamless interaction between the physical and digital worlds



## About Us

We are innovators merging AR and OpenCV to redefine interactive visual experiences



### We Are AR Video Visualizer

At AR Video Visualizer, we're passionate about pushing the boundaries of augmented reality technology. Leveraging the power of OpenCV-Python, we aim to create immersive experiences where digital content seamlessly integrates with the real world. Our team is dedicated to developing innovative solutions for dynamic QR code generation, real-time video rendering, and precise object recognition.

Join us on our journey to redefine visual interaction and explore new frontiers in augmented reality

## Why Choose Us



### Cutting-Edge Technology

We harness the latest advancements in OpenCV-Python and augmented reality to deliver state-of-the-art solutions that redefine how we interact with digital content



### Seamless Integration

Our project seamlessly integrates digital elements into the physical world, providing users with immersive and interactive experiences that blur the lines between reality and imagination



### Reliable Performance

With our precise object recognition and real-time video rendering capabilities, users can trust that our system will consistently deliver high-quality results, ensuring a smooth and engaging experience every time



### Innovation and Creativity

We're not just building a project; we're pushing the boundaries of what's possible in augmented reality. By choosing us, users join a community dedicated to innovation, creativity, and exploring new frontiers in visual interaction

### Address

📍 Chitkara University  
☎ Call +91 82888-51361  
✉ ARVideoVisualizer@gmail.com



### Links

[Home](#)  
[About](#)  
[Services](#)  
[Why Us](#)  
[Model](#)

### Subscribe

Enter email

Subscribe

© All Rights Reserved By AR Video Visualizer



## 9. Conclusion and Future Scope

In conclusion, the successful integration of Paper Vision augmented reality with advanced machine learning techniques has resulted in a powerful and immersive AR application. Through rigorous data preprocessing, model training, and AR environment development, we have achieved real-time object recognition, marker less tracking, and dynamic content integration. The optimized ML models have ensured high accuracy and responsiveness, enhancing user engagement and interaction within the AR environment. Moving forward, the future scope of this project encompasses several exciting avenues.

One key aspect is the continuous refinement and enhancement of ML models to handle more complex object interactions, environmental understanding, and context-aware content placement. Exploring diverse applications in education, such as interactive learning experiences and virtual laboratories, holds immense potential for transforming traditional learning methods. Additionally, integrating AR into training simulations for industries like healthcare, manufacturing, and automotive can improve efficiency, safety, and skill development.

Scalability remains a critical focus, with optimizations for edge computing, cloud deployment, and cross-platform compatibility to ensure widespread accessibility and seamless user experiences across devices. Collaboration with industry partners, academia, and the AR/ML community will drive innovation, foster knowledge sharing, and contribute to the evolution of immersive technologies. Overall, this project lays a strong foundation for advancing ML-driven AR applications, pushing the boundaries of digital experiences and paving the way for innovative solutions in diverse domains.

## 10. References

1. <https://www.mdpi.com/2076-3417/12/10/5159>
2. [\(PDF\) Augmented Reality and Machine Learning Incorporation Using YOLOv3 and ARKit \(researchgate.net\)](#)
3. <https://blog.roboflow.com/computer-vision-ar-experiences/>
4. <https://www.geeksforgeeks.org/face-recognition-using-artificial-intelligence/>
5. <https://pyimagesearch.com/2021/01/04/opencv-augmented-reality-ar/>
6. <https://github.com/topics/augmented-reality>