A Mini Project Report

On

**VIRTUAL MOUSE USING AI**

By

**SIRIKONDA SAI AJEETH**

**(1602-20-733-097);**

**&**

**NAGELLI TARUN KUMAR**

**(1602-20-733-114)**

****

**Department of Computer Science & Engineering**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2021**

**DECLARATION BY THE CANDIDATE**

I , **S.Sai Ajeeth** bearing hall ticket number,**1602-20-733-097,** hereby declare that the project report entitled

“VIRTUAL MOUSE USING AI”

Department of Computer Science & Engineering, VCE, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering**.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**S.Sai Ajeeth**

**1602-20-733-097**

**DECLARATION BY THE CANDIDATE**

I , **N.Tarun Kumar** bearing hall ticket number,

**1602-20-733-114,** hereby declare that the project report entitled

“VIRTUAL MOUSE USING AI”

Department of Computer Science & Engineering, VCE, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering**.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**Nagelli Tarun Kumar**

**1602-20-733-114**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Computer Science & Engineering**

****

**BONAFIDE CERTIFICATE**

This is to certify that the project entitled **“VIRTUAL MOUSE USING AI”** being submitted by bearing 1602-20-733-097,1602-20-733-114**,** in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by him/her under my guidance.

**Dr. T. Adilakshmi,**

**Professor & HOD,**

**Dept. of CSE,**

**TABLE OF CONTENTS**

1.ACKNOWLEDGEMENT…………………… 1

2.PROJECT DESCRIPTION………………... 2

3.ABSTRACT…………………………………..3

4.INTRODUCTION…………………………….4

5.REQUIREMENTS…………………………...5

6.SOURCE CODE.........................................6

7.RESULTS/OUTPUT SCREENSHOTS……21

8.LIMITATIONS ………………………………25

9.FUTURE WORK…....................................25

10.CONCLUSION…………………………… 26

11.REFERENCES…………………………......26

**LIST OF FIGURES**

Fig1 Co-ordinates or land marks in the hand  …..18

Fig2 MediaPipe hand recognition graph…..19

Fig3 Flowchart of the real-time AI virtual mouse system...20

**ACKNOWLEDGEMENT**

The enduring pages of the work are the cumulative sequence of extensive guidance and arduous work. We wish to acknowledge and express our personal gratitude to all those without whom this work could not have been reality.

We feel very delighted to get this rare opportunity to show our profound senses of reverences and indebtedness to our esteemed lecturers [**S. Komal Kour**, Assistant professor](https://www.vce.ac.in/Faculty_Details.cshtml?id=2045) and [**P.Samyuktha**, Assistant professor](https://www.vce.ac.in/Faculty_Details.cshtml?id=2121) , for their keen and sustained interest, valuable advice, throughout the course of which led our mini project, to a successful completion. For this kind act of consideration, we are beholder to them in special manner and no one can fully convey our feelings of respect and regard for them.

1

**PROJECT DESCRIPTION**

Over the recent years, computer vision has started to play a significant role in the Human Computer Interaction (HCI). • With efficient object tracking algorithms, it is possible to track the motion of a human hand in real time using a simple web camera. This presentation discusses the design of a system that tracks the fingertip of the index finger for the purpose of controlling mouse pointer on the screen.

 A single camera (web camera) is used to track the motion of the fingertip in real time. The camera is mounted on the top of the computer monitor or hooked on the laptop screen.

2

**ABSTRACT**

This project presents a new approach for controlling mouse movement using a real-time camera. Major approaches consist of adding more buttons or changing the position of the tracking ball of mouse. Instead, we suggest to change the design of hardware. Our concept is to use a camera and computer vision technology, as image segmentation and gesture recognition, to control mouse tasks (clicking and scrolling) and we show how it can perform everything current mouse devices can. This project shows how to build this mouse control system.

3

**INTRODUCTION**

Since the computer technology continues to grow up, the importance of human computer interaction is enormously increasing. Nowadays most of the mobile devices are using a touch screen technology. However, this technology is still not cheap enough to be used in desktop systems. Creating a virtual human computer interaction device such as mouse or keyboard using a webcam and computer vision techniques can be an alternative way for the touch screen. In this study, finger tracking based a virtual mouse application has been designed and implemented using a regular webcam. The motivation was to create an object tracking application to interact with the computer, and develop a virtual human computer interaction device.

4

**REQUIREMENTS**

**Hardware Requirements**

* 512 MB RAM
* 2GB HDD
* WEBCAM

**Software Requirements**

* Windows XP/ Windows7/8/10/11
* PYTHON INTERPRETER

5

**SOURCE CODE:** CONTAINS TWO MODULES

1.HAND TRACKING MODULE

2.MAIN MODULE

***HAND TRACKING MODULE***

import math

import mediapipe as mp

import numpy as np

import cv2.cv2 as cv

mp\_drawing = mp.solutions.drawing\_utils

mp\_hands = mp.solutions.hands

tipIds = [4,8,12,16,20]

def Hand\_land\_marks(image,hands):

image.flags.writeable = False

result = hands.process(image)

image.flags.writeable = True

6

print(result)

if result.multi\_hand\_landmarks:

for num, hand in

enumerate(result.multi\_hand\_landmarks):

mp\_drawing.draw\_landmarks(image, hand, mp\_hands.HAND\_CONNECTIONS)

def findPosition(image,hands, handNo=0, draw=True):

xList = []

yList = []

bbox = []

lmList = []

result = hands.process(image)

if result.multi\_hand\_landmarks:

myHand = result.multi\_hand\_landmarks[handNo]

7

for id , lm in enumerate(myHand.landmark):

h,w,c=image.shape

cx,cy = int(lm.x\*w),int(lm.y\*h)

xList.append(cx)

yList.append(cy)

lmList.append([id,cx,cy])

if draw:

cv.circle(image,(cx,cy),5,(155,0,155),cv.FILLED)

xmin,xmax = min(xList),max(xList)

ymin,ymax = min(yList),max(yList)

bbox = xmin,ymin,xmax,ymax

if draw:

8

cv.rectangle(image,(xmin - 20, ymin - 20), (xmax + 20, ymax + 20),(0, 255, 0), 2)

return lmList

def fingerup(lmList):

fingers = []

if lmList[tipIds[0]][1] > lmList[tipIds[0]-1][1]:

fingers.append(1)

else:

fingers.append(0)

for id in range(1,5):

if lmList[tipIds[id]][2] < lmList[tipIds[id]-2][2]:

fingers.append(1)

9

else:

fingers.append(0)

return fingers

def findDistance(lmList,p1 , p2 , image , draw = True , r =15 ,t=3):

x1,y1 = lmList[p1][1:]

x2, y2 = lmList[p2][1:]

cx , cy = (x1+x2)//2,(y1+y2)//2

if draw:

cv.line(image,(x1,y1),(x2,y2),(255,0,255),t)

cv.circle(image,(x1,y1),r,(255,0,255),cv.FILLED)

cv.circle(image,(x2,y2),r,(255,0,255),cv.FILLED)

cv.circle(image, (cx,cy), r, (0, 0, 255), cv.FILLED)

10

length = math.hypot(x2 - x1 , y2 - y1)

return length , image,[x1,y1,x2,y2,cx,cy]

11

***MAIN MODULE***

import numpy as np

import handMarking as hm

import cv2.cv2 as cv

import mediapipe as mp

import autopy

import pyautogui as pag

wCam , hCam = 640,480

frameR = 100

smoothening = 7

pTime = 0

plocX, plocY=0, 0

clocX, clocY=0,0

mp\_hands = mp.solutions.hands

12

cap = cv.VideoCapture(0)

cap.set(3, wCam)

cap.set(4,hCam)

wScr , hScr = autopy.screen.size()

lmlist = []

with mp\_hands.Hands(min\_detection\_confidence=0.8,

min\_tracking\_confidence=0.5, max\_num\_hands=1) as

hands:

while cap.isOpened():

sucess, frame = cap.read()

image = cv.cvtColor(frame, cv.COLOR\_RGB2BGR)

#step-1 to mark hand land marks

hm.Hand\_land\_marks(image,hands)

lmlist = hm.findPosition(image,hands)

13

try:

if len(lmlist) != 0:

x1,y1 = lmlist[8][1:]

x2,y2 = lmlist[12][1:]

else :

pass

fingers = hm.fingerup(lmlist)

cv.rectangle(image,(frameR,frameR),(wCam-

frameR,hCam-frameR),(255,0,255),2)

except:

pass

try:

if fingers[1] == 1 and fingers[2]==0:

14

x3 = np.interp(x1,(frameR,wCam-frameR),(0,wScr))

y3 = np.interp(y1, (frameR,hCam-frameR),(0,hScr))

clocX = plocX + (x3-plocX) / smoothening

clocY = plocY + (y3-plocY)/smoothening

autopy.mouse.move(wScr-clocX,clocY)

cv.circle(image,(x1,y1),15,(255,0,255),cv.FILLED)

plocX,plocY = clocX,clocY

except:

pass

try:

if fingers[1] == 1 and fingers[2] == 1 and fingers[0] == 0:

length,image,lineInfo =

hm.findDistance(lmlist,8,12,image)

15

print(length)

if length <40:

cv.circle(image,(lineInfo[4],lineInfo[5]),15,(0,255,0),cv.FILLED)

pag.click(button='left', clicks=3, interval=0.25)

except:

pass

try:

if fingers[1] == 1 and fingers[2] == 1 and fingers[0] == 1:

length,image,lineInfo = hm.findDistance(lmlist,8,12,image)

print(length)

if length<40:

cv.circle(image,(lineInfo[4],lineInfo[5]),15,(0,225,0),cv.FILLED)

pag.click(button='right', clicks=3, interval=0.25)

except:

16

pass

cv.imshow("Ai mouse",image)

cv.waitKey(1)

17

Co-ordinates or land marks in the hand

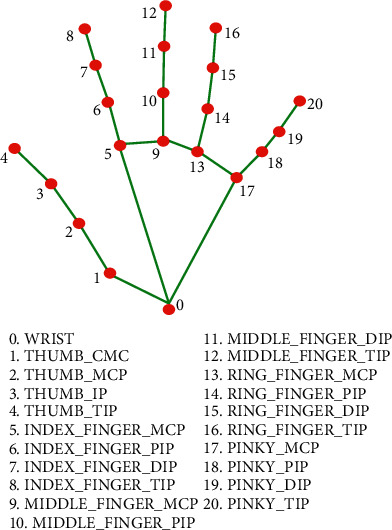
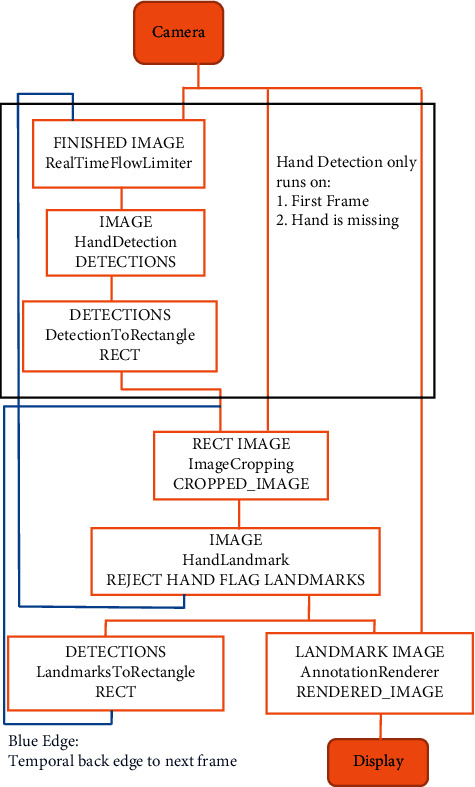
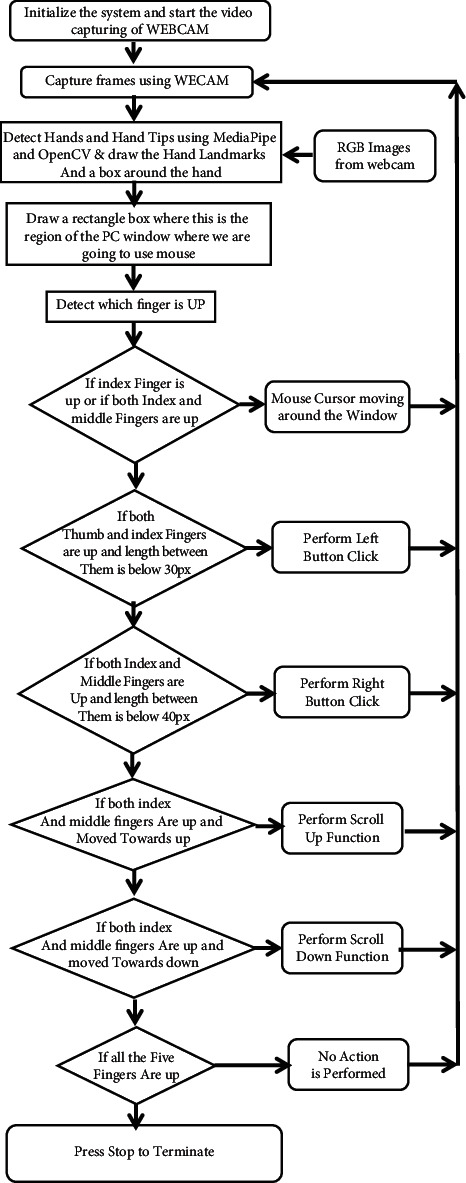
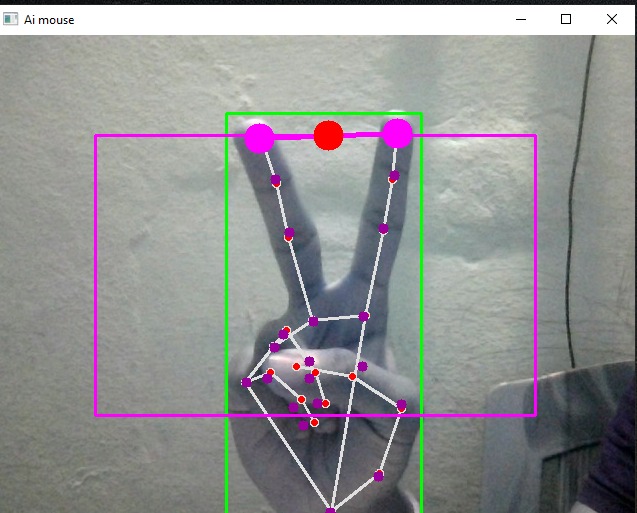


Fig 1. Co-ordinates or land marks in the hand

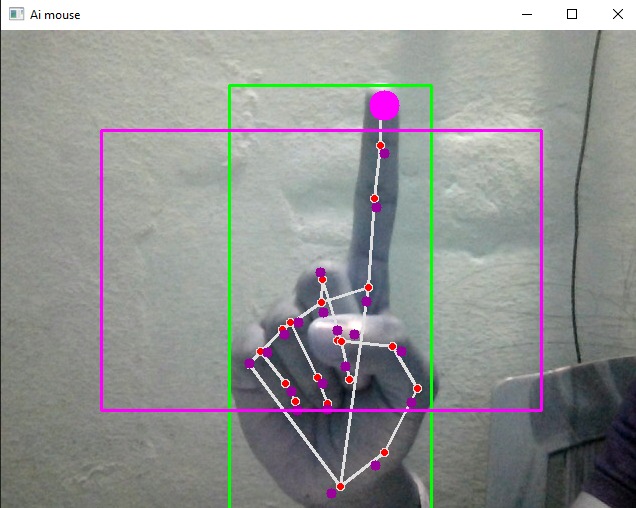
18

19

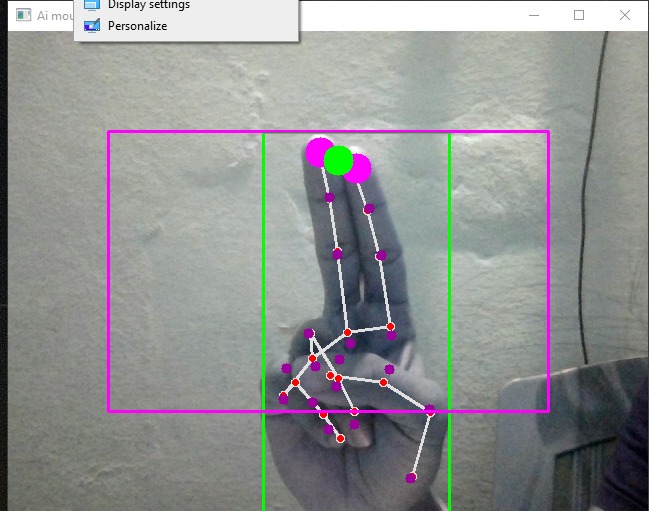
20OUTPUT EXECUTION SCREENSHOTS :



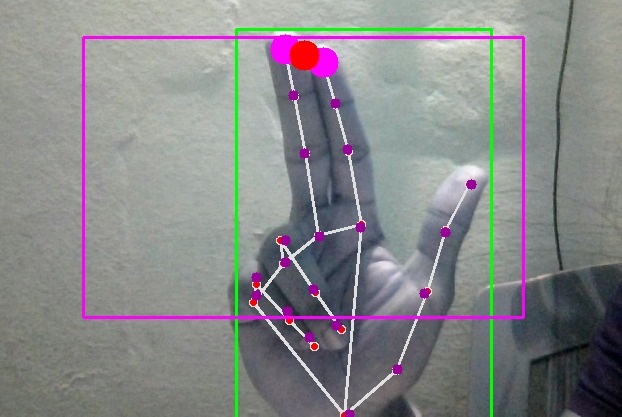
*Activating click*

 *Cursor Moving mode*

21



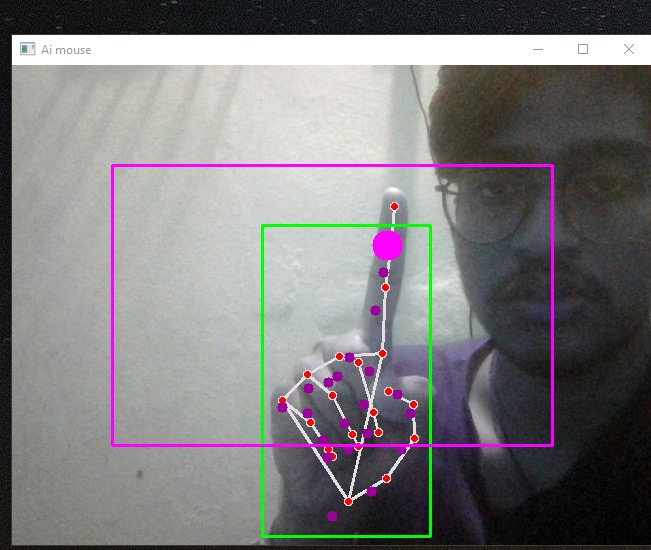
*Right click*

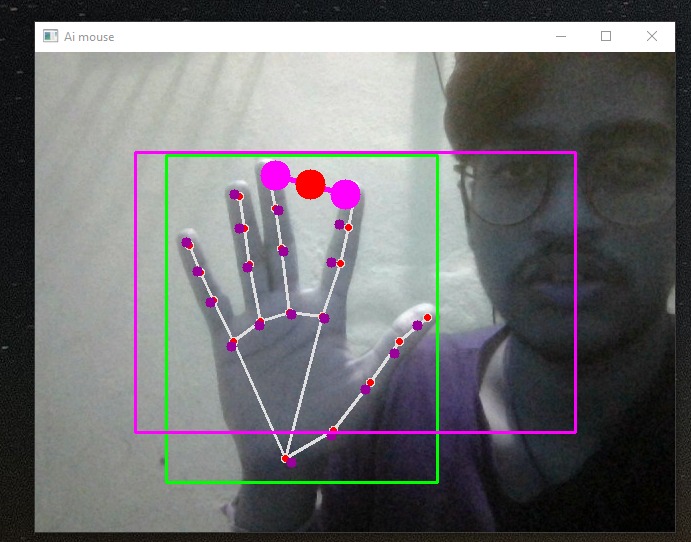


*Left*

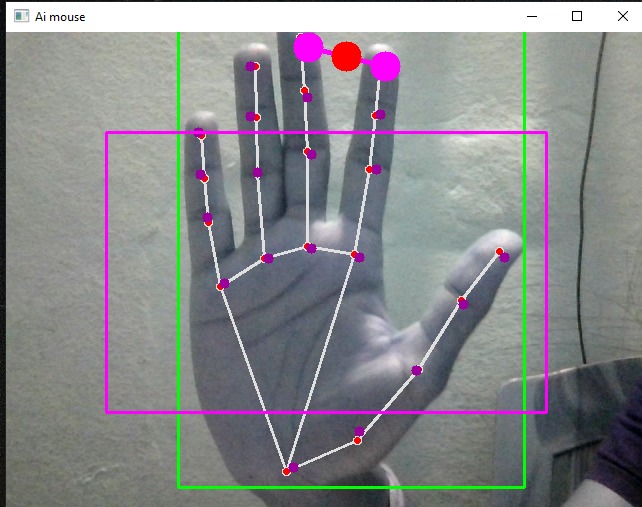
*click*

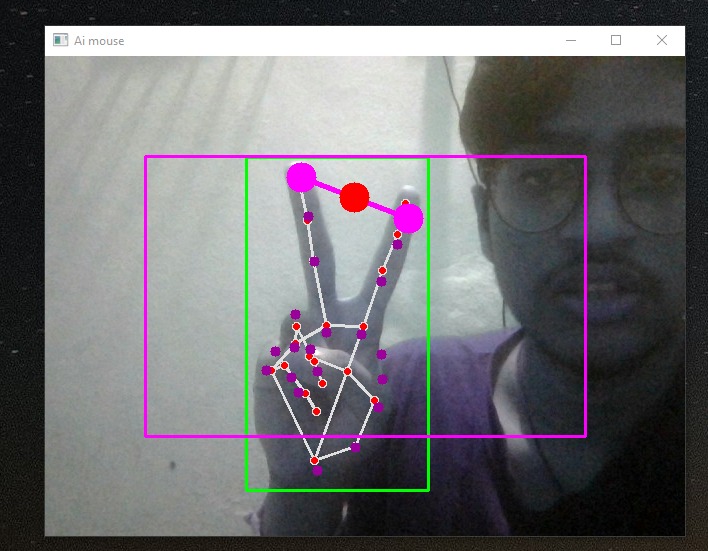
22





23





24

[**LIMITATIONS**](https://image1.slideserve.com/2392016/slide24-l.jpg)

• If background contains colors similar to skin, then the algorithm will loose track of the hand or falsely report its location.

• When the camera's height is changed, the system has reported false pose detection. A better way to detect pointing pose is to use Machine learning algorithm (for example Neural network ).

• The mouse cursor movement on the screen required more smoothing. Also user is not able to cover the entire screen.

**FUTURE WORK**

In the future, we plan to add more features such as enlarging and shrinking windows, smoothening cursoretc. by using the palm and multiple fingers. • We can also open the browser or any drives (C: /D:/E: etc)with the help of hand gestures instead of moving the cursor.

We also can add virtual keyboard and we can use both mouse and keyboard with our hand gestures but not both simultaneously.

25

**CONCLUSION**

We are developing a system to control the mouse cursor using a real-time camera.

• This system is based on computer vision algorithms and can do all mouse tasks.

• However, it is difficult to get stable results because of the variety of lighting and skin colors of human races.

• This system could be useful in presentations and to reduce work space.

• Features such as enlarging and shrinking windows, closing window, etc. by using the palm and multiple fingers.

**REFERENCES**

<https://www.computervision.zone/courses/ai-virtual-mouse/>

<https://pythonrepo.com/repo/HxnDev-Virtual-Mouse-using-OpenCV-python-computer-vision>

<https://pubmed.ncbi.nlm.nih.gov/34733459/>

<http://www.ijarset.com/upload/2021/april/17-bhegadeneha-15.PDF>

Katona J. A review of human–computer interaction and virtual reality research fields in cognitive InfoCommunications. Applied Sciences . 2021;11(6):p. 2646. doi: 10.3390/app11062646. - [DOI](https://doi.org/10.3390/app11062646)

26