CAT 3 PROJECT

LIVE VIDEO TRANSMISSION

Problem Statement:

In our day to day life communication plays an vital role, as its evolution chatting and video conferencing blooms in the current era. As our project deals with this, we have implemented an live video transmission with a chat access concurrently. We used cv2, the advanced package of cv which supports us for video transmission, whereas cv won't support video transmission. We included message access for the user. The server and client will connect through same IP address, so they can connect through the video call and with the chat access.

Code:

Server.py

```
import cv2, imutils, socket
import numpy as np
import time
import base64
import threading, wave, pyaudio,pickle,struct
import sys
import queue
import os
q = queue.Queue(maxsize=10)



BUFF_SIZE = 65536
server_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
server_socket.setsockopt(socket.SOL_SOCKET,socket.SO_RCVBUF,BUFF_SIZE)
host_name = socket.gethostname()
host_ip = '192.168.170.58'#  socket.gethostbyname(host_name)
print(host_ip)
port = 9699
socket_address = (host_ip,port)
server_socket.bind(socket_address)
print('Listening at:',socket_address)

vid = cv2.VideoCapture(0)


def generate_video():

    WIDTH=400
```

```
    while(vid.isOpened()):
        try:
            _,frame = vid.read()
            frame = imutils.resize(frame,width=WIDTH)
            q.put(frame)
        except:
            os._exit(1)
        time.sleep(0.001)
    print('Player closed')
    BREAK=True
    vid.release()

def send_video():
    fps,st,frames_to_count,cnt = (0,0,20,0)
    cv2.namedWindow('SERVER TRANSMITTING VIDEO')
    cv2.moveWindow('SERVER TRANSMITTING VIDEO', 400,30)
    # while True:
    msg,client_addr = server_socket.recvfrom(BUFF_SIZE)
    print('GOT connection from ',client_addr)
    WIDTH=400
    while(True):

        frame = q.get()
        encoded,buffer = cv2.imencode('.jpeg',frame,[cv2.IMWRITE_JPEG_QUALITY,80])
        message = base64.b64encode(buffer)
        server_socket.sendto(message,client_addr)
        frame = cv2.putText(frame,'FPS:
'+str(round(fps,1)),(10,40),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)

        if cnt == frames_to_count:
            try:
                fps = round(frames_to_count/(time.time()-st),1)
                st=time.time()
                cnt=0
            except:
                pass
        cnt+=1

        cv2.imshow('SERVER TRANSMITTING VIDEO', frame)
        key = cv2.waitKey(1) & 0xFF
        if key == ord('q'):
            os._exit(1)
        time.sleep(0.01)

def send_message():
    s = socket.socket()
    s.bind((host_ip, (port-1)))
    s.listen(5)
    client_socket,addr = s.accept()
    cnt=0
```

```
    while True:
        if client_socket:
            while True:
                print('SERVER TEXT ENTER BELOW:')
                data = input()
                data=data+' ('+str(len(data))+')'
                a = pickle.dumps(data)
                message = struct.pack("Q",len(a))+a
                client_socket.sendall(message)

                cnt+=1
                time.sleep(0.01)



def get_message():
    s = socket.socket()
    s.bind((host_ip, (port-2)))
    s.listen(5)
    client_socket,addr = s.accept()
    data = b""
    payload_size = struct.calcsize("Q")

    while True:
        try:
            while len(data) < payload_size:
                packet = client_socket.recv(4*1024) # 4K
                if not packet: break
                data+=packet
            packed_msg_size = data[:payload_size]
            data = data[payload_size:]
            msg_size = struct.unpack("Q",packed_msg_size)[0]
            while len(data) < msg_size:
                data += client_socket.recv(4*1024)
            frame_data = data[:msg_size]
            data  = data[msg_size:]
            frame = pickle.loads(frame_data)
            print('',end='\n')
            print('CLIENT TEXT RECEIVED:',frame,end='\n')
            print('SERVER TEXT ENTER BELOW:')
            time.sleep(0.001)

        except Exception as e:
            print('Dropped...')
            pass

    client_socket.close()
    print('Audio closed')
```

```
def get_video():

    cv2.namedWindow('SERVER RECEIVING VIDEO')
    cv2.moveWindow('SERVER RECEIVING VIDEO', 400,360)
    fps,st,frames_to_count,cnt = (0,0,20,0)
    BUFF_SIZE = 65536
    server_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
    server_socket.setsockopt(socket.SOL_SOCKET,socket.SO_RCVBUF,BUFF_SIZE)
    socket_address = (host_ip,port-3)
    server_socket.bind(socket_address)
    while True:

        packet,_ = server_socket.recvfrom(BUFF_SIZE)
        data = base64.b64decode(packet,' /')
        npdata = np.fromstring(data,dtype=np.uint8)

        frame = cv2.imdecode(npdata,1)
        frame = cv2.putText(frame,'FPS:
'+str(fps),(10,40),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
        cv2.imshow("SERVER RECEIVING VIDEO",frame)
        key = cv2.waitKey(1) & 0xFF

        if key == ord('q'):
            # client_socket.close()
            break


        if cnt == frames_to_count:
            try:
                fps = round(frames_to_count/(time.time()-st),1)
                st=time.time()
                cnt=0
            except:
                pass
        cnt+=1
        time.sleep(0.001)

    # client_socket.close()
    cv2.destroyAllWindows()



t1 = threading.Thread(target=send_message, args=())
t2 = threading.Thread(target=get_message, args=())
t3 = threading.Thread(target=generate_video, args=())
t4 = threading.Thread(target=send_video, args=())
t5 = threading.Thread(target=get_video, args=())
t1.start()
```

```
t2.start()
t3.start()
t4.start()
t5.start()
```

Client.py

```
import cv2, imutils, socket
import numpy as np
import time, os
import base64
import queue
import threading, wave, pyaudio,pickle,struct
# For details visit pyshine.com
BUFF_SIZE = 65536

BREAK = False
client_socket = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
client_socket.setsockopt(socket.SOL_SOCKET,socket.SO_RCVBUF,BUFF_SIZE)
host_name = socket.gethostname()
host_ip = '192.168.170.58'#  socket.gethostbyname(host_name)
print(host_ip)
port = 9699
message = b'Hello'

client_socket.sendto(message,(host_ip,port))


q = queue.Queue(maxsize=10)
vid = cv2.VideoCapture(1)


def generate_video():

    WIDTH=400
    while(vid.isOpened()):
        try:
            _,frame = vid.read()
            frame = imutils.resize(frame,width=WIDTH)
            q.put(frame)

        except:
            os._exit(1)
        time.sleep(0.001)
    print('Player closed')
    BREAK=True
    vid.release()

def get_message():
```

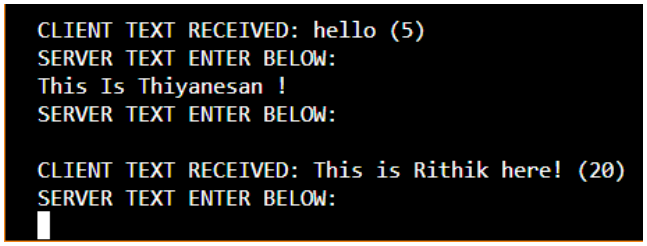
```
    # TCP socket
    client_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    socket_address = (host_ip,port-1)
    print('server listening at',socket_address)
    client_socket.connect(socket_address)
    print("CLIENT CONNECTED TO",socket_address)
    data = b""
    payload_size = struct.calcsize("Q")
    while True:
        try:
            while len(data) < payload_size:
                packet = client_socket.recv(4*1024) # 4K
                if not packet: break
                data+=packet
            packed_msg_size = data[:payload_size]
            data = data[payload_size:]
            msg_size = struct.unpack("Q",packed_msg_size)[0]
            while len(data) < msg_size:
                data += client_socket.recv(4*1024)
            frame_data = data[:msg_size]
            data  = data[msg_size:]
            frame = pickle.loads(frame_data)
            print('',end='\n')
            print('SERVER TEXT RECEIVED:',frame,end='\n')
            print('CLIENT TEXT ENTER BELOW:')
            time.sleep(0.01)
        except:

            break

    client_socket.close()
    print('closed')
    os._exit(1)


def send_message():

    # create socket
    client_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    socket_address = (host_ip,port-2)
    print('server listening at',socket_address)
    client_socket.connect(socket_address)
    print("msg send CLIENT CONNECTED TO",socket_address)
    while True:
        if client_socket:
            while (True):
                print('CLIENT TEXT ENTER BELOW:')
                data = input ()
                data=data+' ('+str(len(data))+')'
```

```
                a = pickle.dumps(data)
                message = struct.pack("Q",len(a))+a
                client_socket.sendall(message)
                time.sleep(0.01)




def get_video():

    cv2.namedWindow('CLIENT RECEIVING VIDEO')
    cv2.moveWindow('CLIENT RECEIVING VIDEO', 10,360)
    fps,st,frames_to_count,cnt = (0,0,20,0)
    message = b'Hello'
    client_socket.sendto(message,(host_ip,port))
    while True:
        packet,_ = client_socket.recvfrom(BUFF_SIZE)
        data = base64.b64decode(packet,' /')
        npdata = np.fromstring(data,dtype=np.uint8)

        frame = cv2.imdecode(npdata,1)
        frame = cv2.putText(frame,'FPS:
'+str(fps),(10,40),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)
        cv2.imshow("CLIENT RECEIVING VIDEO",frame)
        key = cv2.waitKey(1) & 0xFF

        if key == ord('q'):
            client_socket.close()
            os._exit(1)


        if cnt == frames_to_count:
            try:
                fps = round(frames_to_count/(time.time()-st),1)
                st=time.time()
                cnt=0
            except:
                pass
        cnt+=1
        time.sleep(0.001)


    client_socket.close()
    cv2.destroyAllWindows()




def send_video():
    socket_address = (host_ip,port-3)
```

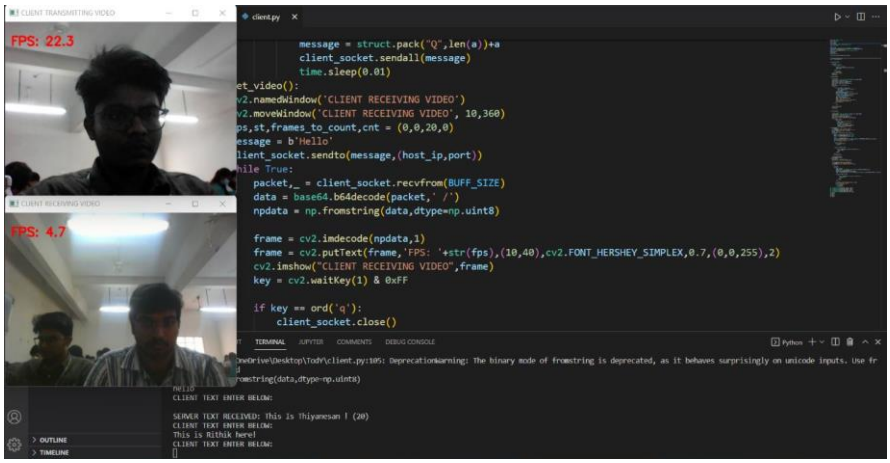
```
    print('server listening at',socket_address)

    fps,st,frames_to_count,cnt = (0,0,20,0)
    cv2.namedWindow('CLIENT TRANSMITTING VIDEO')
    cv2.moveWindow('CLIENT TRANSMITTING VIDEO', 10,30)
    while True:

        WIDTH=400
        while(True):
            frame = q.get()
            encoded,buffer = cv2.imencode('.jpeg',frame,[cv2.IMWRITE_JPEG_QUALITY,80])
            message = base64.b64encode(buffer)
            client_socket.sendto(message,socket_address)
            frame = cv2.putText(frame,'FPS:
'+str(round(fps,1)),(10,40),cv2.FONT_HERSHEY_SIMPLEX,0.7,(0,0,255),2)

            if cnt == frames_to_count:
                try:
                    fps = round(frames_to_count/(time.time()-st),1)
                    st=time.time()
                    cnt=0
                except:
                    pass
            cnt+=1

            cv2.imshow('CLIENT TRANSMITTING VIDEO', frame)
            key = cv2.waitKey(1) & 0xFF
            if key == ord('q'):
                os._exit(1)
            time.sleep(0.001)


t1 = threading.Thread(target=get_message, args=())
t2= threading.Thread(target=send_message, args=())
t3 = threading.Thread(target=get_video, args=())
t4 = threading.Thread(target=send_video, args=())
t5 = threading.Thread(target=generate_video, args=())
t1.start()
t2.start()
t3.start()
t4.start()
t5.start()
```

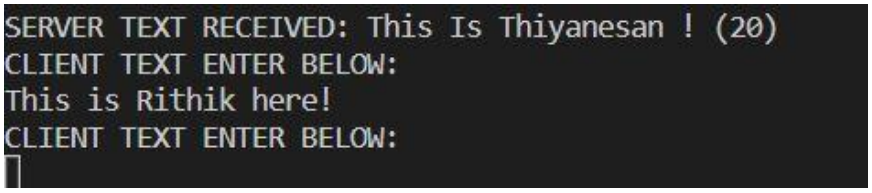
Output:

Server Side



Client Side:




SERVER TEXT RECEIVED: This Is Thiyanesan ! (20)
CLIENT TEXT ENTER BELOW:
This is Rithik here!
CLIENT TEXT ENTER BELOW: