

教育部「5G行動寬頻人才培育跨校教學聯盟計畫」
5G行動網路協定與核網技術聯盟中心

Mobile Edge Computing: 行動邊緣計算

實驗單元-04：AR / VR 邊緣計算實驗 (Off-Loading)

授課教師：萬欽德

授課助教：林鴻章

國立高雄科技大學 電腦與通訊工程系

Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

實驗目的

- 使用 MEC 平台實作 AR/VR 應用。
- 結合影像處理技術進行 AR 視訊處理。

實驗內容*

- 了解Off-Loading實驗流程
- AR / VR 邊緣計算實驗

Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

實驗架構

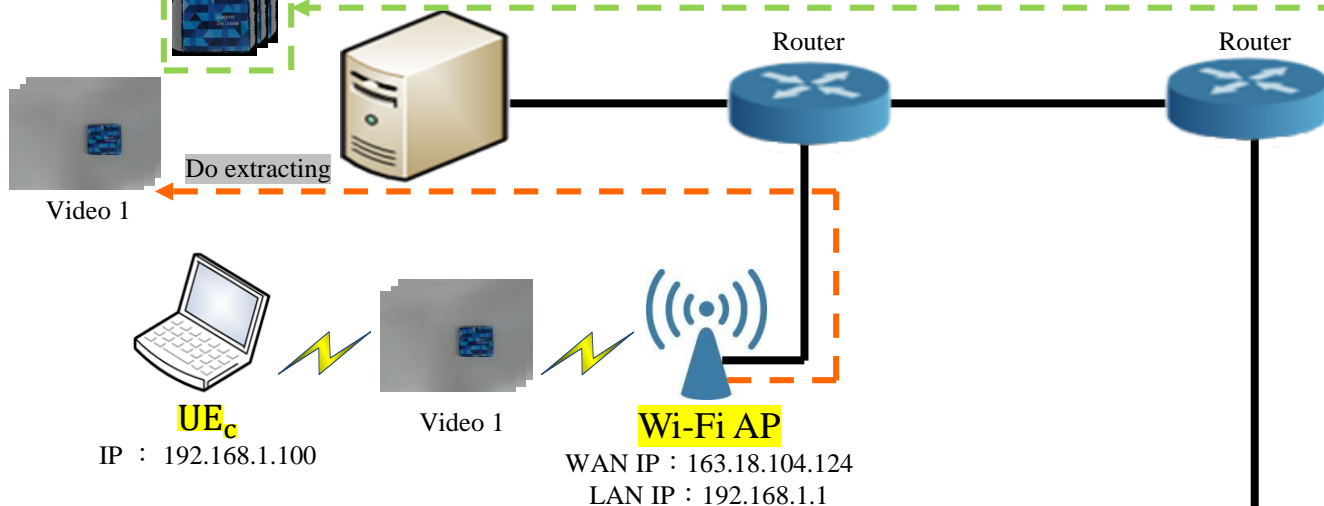
Cloud Network

Extracted Object

Cloud Server

WAN IP : 163.18.104.142

Video 1
Video 2
Server Communication



Edge Network

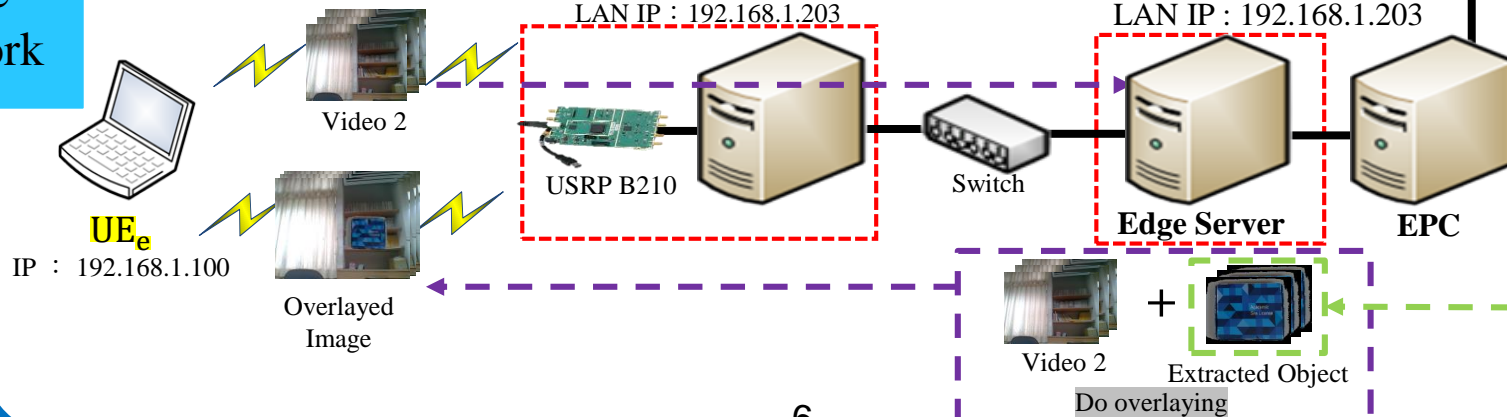
eNB

LAN IP : 192.168.1.203

Edge Server (on EPC)

WAN IP : 163.18.104.132

LAN IP : 192.168.1.203



實驗設備-硬體(1/2)

名稱	硬體	數量	目的
Cloud Server	CPU : i7-6500U RAM : 8 GB HDD : 1 TB	1 台	提取UE _c 的影片後，傳送至 Edge Server
	Ethernet Network PCIE Card	1 個	連接WAN (for Internet)
Edge Server	CPU : i7-8559U RAM : 8 GB HDD : 1 TB	1 台	接收到Cloud Server影片後，跟UE _e 上傳影片做overlay，並上傳至UE _e
	Ethernet Network PCIE Card	1 個	連接WAN (for Internet)
	RJ45外接網卡 (USB)	1 個	連接LAN (for eNB)
eNB	CPU : i7-8559U RAM : 8 GB HDD : 1 TB	1 台	eNodeB基地台
	Ethernet Network PCIE Card	1 個	連接EPC的LAN
	USRP B210	1 片	接收 eNB 封包資料，轉成 LTE 訊號發送
	VERT2450 Antenna	2 支	收發 LTE Band 7 (2600 MHz) 訊號
	USB 3.0 cable	1 條	連接 eNB 與 USRP B210

實驗設備-硬體(2/2)

名稱	硬體	數量	目的
UE _c	CPU : i7-9750H RAM : 8 GB HDD : 256 GB (筆電 具備攝像頭)	1 台	上傳影片至Cloud Server
UE _e	CPU : i5-6200U RAM : 4 GB HDD : 500 GB (筆電 具備攝像頭)	1 台	連接eNB 上傳影片至Edge Server 下載overlay後的影片
	4G Dongle	1個	提供電腦使用行動網路
	LTE SIM卡	1張	提供UE使用，註冊EPC
Wi-Fi AP	D-Link Wireless N 8	1 台	提供UE _c 網路
Switch	4-port switch (legacy)	1 台	LAN互相連接
RJ45雙絞線	RJ45雙邊接頭的CAT 5e網路線	5條	Edge*2條、eNB*1條 Cloud*1條、Wi-Fi AP*1條

實驗設備-軟體

名稱	軟體	版本
Cloud Server	OS : Ubuntu	16.04 LTS
	Nginx	1.5.0
	OpenCV	4.1.0
Edge Server	OS : Ubuntu	16.04 LTS
	OAI-EPC	https://gitlab.eurecom.fr/oai/openair-cn.git (發布日期:2017/3/31)
	Nginx	1.5.0
	OpenCV	4.1.0
eNB	OS : Ubuntu	16.04 LTS
	OAI-eNB	https://gitlab.eurecom.fr/oai/openairinterface5g/tree/17b9a9e917ce2a3a8c7004c7b9a221c350ddfe17 (發布日期:2015/8/8)
UE _c	OS : Ubuntu	16.04 LTS
	FFmpeg	2.8.17
UE _e	OS : Ubuntu	16.04 LTS
	FFmpeg	2.8.17

Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

Cloud Server 安裝需求

- Cloud Server的安裝需求(ubuntu 16.04)
 - 1.Video Streaming Server 安裝
 - Nginx 軟體，請參考實驗單元-02
 2. Digital Image Processing 安裝
 - OpenCV 軟體，請參考實驗單元-02
 3. TCP/UDP Socket 安裝
 - C Language 軟體，請參考實驗單元-02
 4. Cloud程式碼
 - 程式碼 請參考附錄下載

Edge Server 安裝需求

- Edge Server的安裝需求(ubuntu 16.04)
 - 1.Video Streaming Server 安裝
 - Nginx 軟體，請參考實驗單元-02
 2. Digital Image Processing 安裝
 - OpenCV 軟體，請參考實驗單元-02
 3. TCP/UDP Socket 安裝
 - C Language 軟體，請參考實驗單元-02
 4. Radio Access Network 安裝
 - OAI-EPC 軟體，請參考實驗單元-02
 5. Edge程式碼
 - 程式碼 請參考附錄下載

UE的安裝需求

- Edge UE的安裝需求(ubuntu 16.04)
 - 1.FFmpeg 安裝
 - FFmpeg 軟體，請參考實驗單元-02
- Cloud UE的安裝需求(ubuntu 16.04)
 - 1. FFmpeg 安裝
 - FFmpeg 軟體，請參考實驗單元-02

Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

Cloud Server 程式修改-1

- 下載Cloud.zip並解壓縮
- <https://drive.google.com/drive/folders/1uljnvdo9FkSSTqVSuCYH45y3OyDDl4qH?usp=sharing>
- 修改Cloud.cpp
- 22行修改成Edge Server IP

```
~
7 #include <time.h>
8 #include <iostream>
9 #include <stdio.h>
10 #include <unistd.h>
11 #include <stdlib.h>
12 #include <time.h>
13 #include <vector>
14
15 #include <string.h>
16 #include <sys/types.h>
17 #include <sys/socket.h>
18 #include <netinet/in.h>
19 #include <netdb.h>
20 #include <sys/time.h>
21
22 #define SERVER_IP "163.18.104.132" /Edge Server IP
23 #define PORT 7200
24 #define FRAME_WIDTH 640
25 #define FRAME_HEIGHT 480
26 #define t_d 1
```

Cloud Server 程式修改-2

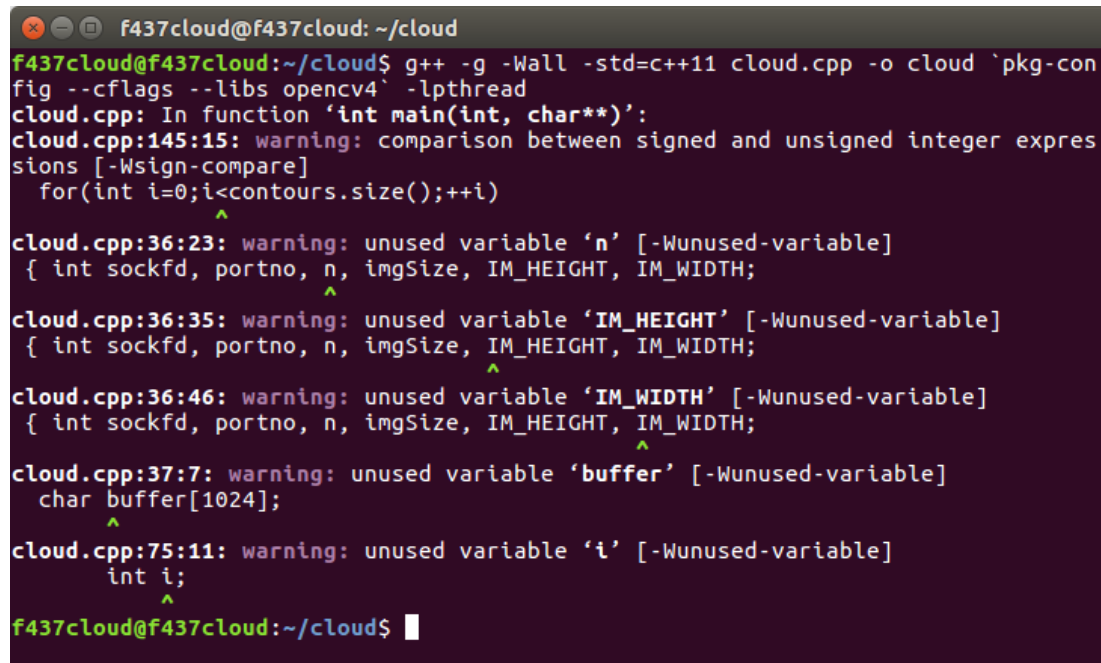
- 修改Cloud.cpp
- 77行修改成Cloud Server IP

```
69 VideoCapture capture,capture_bk;
70
71 vector<vector<Point> > contours;
72 vector<Vec4i> hierarchy;
73
74     Mat image,im_in,im_in1,im_d,image_bk,mask_t,cameraFeed,background;
75     int i;
76     //capture.open(0);
77     const std::string videoStreamAddress = "rtmp://163.18.104.142:1935/myapp/test"; //Cloud
    UE frame (object)
78
79
80     //const std::string videoStreamAddress2 = "test.mp4";
81
82 if((!capture.open(videoStreamAddress))) {
83     std::cout << "Error opening video stream or file" << std::endl;
84     return -1;
85 }
86 /* if((!capture_bk.open(videoStreamAddress2))) {
87     std::cout << "Error opening video stream or file" << std::endl;
88     return -1;
```


Cloud Server 程式編譯

編譯Cloud.cpp程式

- `$ sudo g++ -g -Wall -std=c++11 Cloud.cpp -o cloud `pkg-config --cflags --libs opencv4` -lpthread`



```
f437cloud@f437cloud: ~/cloud
f437cloud@f437cloud:~/cloud$ g++ -g -Wall -std=c++11 cloud.cpp -o cloud `pkg-config --cflags --libs opencv4` -lpthread
cloud.cpp: In function 'int main(int, char**)':
cloud.cpp:145:15: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
    for(int i=0;i<contours.size();++i)
                  ^
cloud.cpp:36:23: warning: unused variable 'n' [-Wunused-variable]
    { int sockfd, portno, n, imgSize, IM_HEIGHT, IM_WIDTH;
                          ^
cloud.cpp:36:35: warning: unused variable 'IM_HEIGHT' [-Wunused-variable]
    { int sockfd, portno, n, imgSize, IM_HEIGHT, IM_WIDTH;
                                   ^
cloud.cpp:36:46: warning: unused variable 'IM_WIDTH' [-Wunused-variable]
    { int sockfd, portno, n, imgSize, IM_HEIGHT, IM_WIDTH;
                                              ^
cloud.cpp:37:7: warning: unused variable 'buffer' [-Wunused-variable]
    char buffer[1024];
        ^
cloud.cpp:75:11: warning: unused variable 'i' [-Wunused-variable]
    int i;
        ^
f437cloud@f437cloud:~/cloud$
```

Edge Server 程式修改-1

- 下載edge.zip並解壓縮
- <https://drive.google.com/drive/folders/1uljnvdo9FkSSTqVSuCYH45y3OyDDl4qH?usp=sharing>
- 修改Edge.cpp
- 18行修改成Edge Server IP

```
5 #include <netinet/in.h>
6 #include <arpa/inet.h>
7 #include <unistd.h>
8 #include <stdlib.h>
9 #include <string.h>
10 #include <sys/time.h>
11 //pthread
12 #include <pthread.h>
13 #include <semaphore.h>
14 #include <time.h>
15
16 #define NUM_THREADS 2
17 #define MSIZE 1024
18 #define SERVER_IP "163.18.104.132" //Edge Server IP
19
20 #define PORT 7200
21 #define BUFF_LEN 1024
22 #define FRAME_WIDTH 640
23 #define FRAME_HEIGHT 480
```

Edge Server 程式修改-2

- 修改Edge.cpp
- 93行修改成Edge Server IP

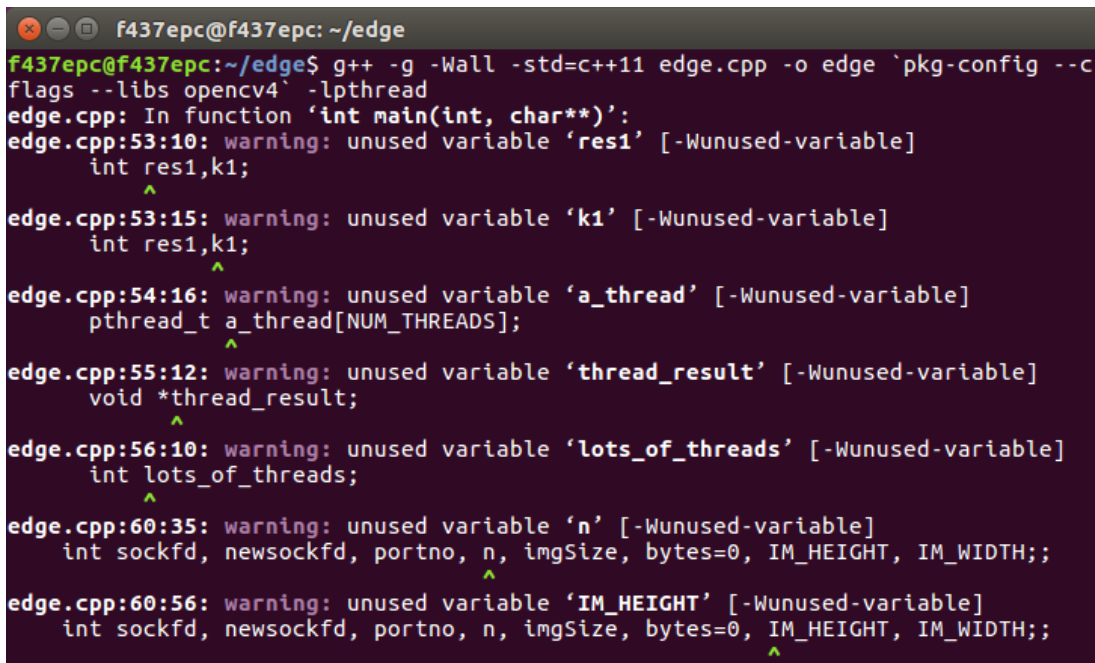
```
89
90 cv::VideoCapture vcap,vcap2;
91 int count=1;
92 int count_t = 1;
93 const std::string videoStreamAddress = "rtmp://163.18.104.132:1935/myapp/test"; //Edge UE
    frame (background)
94 // const std::string videoStreamAddress2 = "cloud.mp4";
95 // const std::string videoStreamAddress2 = "camera1.mp4";
96
97 Mat background;
98
99
100
101
102 if(!vcap.open(videoStreamAddress)) {
103     std::cout << "Error opening video stream or file" << std::endl;
104     return -1;
105 }
```

Edge Server IP

Edge Server 程式編譯

編譯Edge.cpp程式

- `$ sudo g++ -g -Wall -std=c++11 Edge.cpp -o edge `pkg-config --cflags --libs opencv4` -lpthread`



```
f437epc@f437epc: ~/edge
f437epc@f437epc:~/edge$ g++ -g -Wall -std=c++11 edge.cpp -o edge `pkg-config --c
flags --libs opencv4` -lpthread
edge.cpp: In function 'int main(int, char**)':
edge.cpp:53:10: warning: unused variable 'res1' [-Wunused-variable]
    int res1,k1;
        ^
edge.cpp:53:15: warning: unused variable 'k1' [-Wunused-variable]
    int res1,k1;
              ^
edge.cpp:54:16: warning: unused variable 'a_thread' [-Wunused-variable]
    pthread_t a_thread[NUM_THREADS];
              ^
edge.cpp:55:12: warning: unused variable 'thread_result' [-Wunused-variable]
    void *thread_result;
           ^
edge.cpp:56:10: warning: unused variable 'lots_of_threads' [-Wunused-variable]
    int lots_of_threads;
        ^
edge.cpp:60:35: warning: unused variable 'n' [-Wunused-variable]
    int sockfd, newsockfd, portno, n, imgSize, bytes=0, IM_HEIGHT, IM_WIDTH;;
                                   ^
edge.cpp:60:56: warning: unused variable 'IM_HEIGHT' [-Wunused-variable]
    int sockfd, newsockfd, portno, n, imgSize, bytes=0, IM_HEIGHT, IM_WIDTH;;
                                                         ^
```

Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

實驗流程

- 實驗的五個步驟：

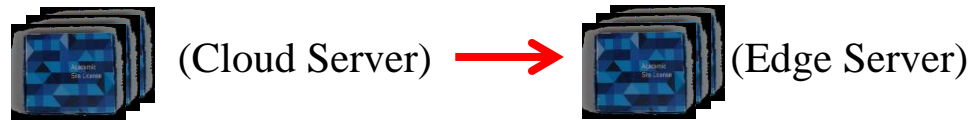
一. Cloud UE將畫面傳送到Cloud Server， Edge UE將畫面傳送到Edge Server



二. Cloud Server將Cloud UE的畫面做擷取



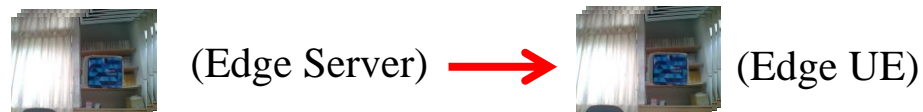
三. Cloud Server傳送(socket)擷取後的物件到Edge Server



四. Edge Server將物件與Edge UE的畫面做重疊



五. Edge UE播放重疊後的畫面



Cloud程式說明-1

1. 建立socket

```
struct sockaddr_in serv_addr;
struct hostent *server;
portno = PORT;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0) error("ERROR opening socket");
server = gethostbyname(SERVER_IP);
if (server == NULL) {
    fprintf(stderr, "ERROR, no such host\n");
    exit(0);
}
bzero((char *) &serv_addr, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
bcopy((char *)server->h_addr,
      (char *)&serv_addr.sin_addr.s_addr,
      server->h_length);
serv_addr.sin_port = htons(portno);
if (connect(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0)
    //error("ERROR connecting");
    return -1;
```

Cloud程式說明-2

2. 得到Cloud UE的畫面

```
Mat image,im_in,im_in1,im_d,image_bk,mask_t,cameraFeed,background;
int i;
//capture.open(0);
const std::string videoStreamAddress = "rtmp://163.18.104.142:1935/myapp/test"; //Cloud UE frame (object)
//const std::string videoStreamAddress2 = "test.mp4";

if(!capture.open(videoStreamAddress)) {
    std::cout << "Error opening video stream or file" << std::endl;
    return -1;
}
```


Cloud程式說明-3

3. 對Cloud UE的畫面做擷取

```
//-----extration_start
cvtColor(image, im_in, CV_RGB2GRAY);
im_in1 = image;
Mat im_th;
threshold(im_in, im_th, 220, 255, THRESH_TRIANGLE);
Mat im_floodfill = im_th.clone();
floodFill(im_floodfill, cv::Point(0,0), Scalar(255));
Mat im_floodfill_inv;
bitwise_not(im_floodfill, im_floodfill_inv); // (not A , B)
imshow("Foregroundgray", im_in1);

Mat im_out = (im_th | im_floodfill_inv );
Mat im_out2 ;

image.copyTo(im_out2, im_floodfill_inv); //C = A & B

findContours(im_floodfill_inv, contours, hierarchy, RETR_EXTERNAL, CHAIN_APPROX_NONE);
vector<vector<Point> > contours1;

for(int i=0; i<contours.size(); ++i)
{
    if(contours[i].size()>200)
    {
        contours1.push_back(contours[i]);
    }
}

Mat mask(image.size(), CV_8U, Scalar(0));
Mat mask_inv;
drawContours(mask, contours, -1, Scalar(255), FILLED);
imshow("mask", mask);

Mat object;
image_bk.copyTo(object, mask);

imshow("object", object);
```

Cloud程式說明-4

4. 傳送擷取後的物件給Edge Server

```
cameraFeed.rows = object.rows;  
cameraFeed.cols = object.cols;  
resize(object, cameraFeed, Size( FRAME_WIDTH, FRAME_HEIGHT ));  
  
imgSize=cameraFeed.total()*cameraFeed.elemSize();  
  
send(sockfd, cameraFeed.data, imgSize, 0);
```

Edge程式說明-1

1. 建立socket

```
int sockfd, newsockfd, portno, n, imgSize, bytes=0, IM_HEIGHT, IM_WIDTH;
socklen_t clilen;
char buffer[2560];
struct sockaddr_in serv_addr, cli_addr;
Mat img;

sockfd=socket(AF_INET, SOCK_STREAM, 0);
if(sockfd<0) error("ERROR opening socket");

bzero((char*)&serv_addr, sizeof(serv_addr));
portno = PORT;

serv_addr.sin_family=AF_INET;
serv_addr.sin_addr.s_addr=inet_addr(SERVER_IP);
serv_addr.sin_port=htons(portno);

if(bind(sockfd, (struct sockaddr *) &serv_addr,
        sizeof(serv_addr))<0) error("ERROR on binding");

listen(sockfd,5);
clilen=sizeof(cli_addr);

newsockfd=accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
if(newsockfd<0) error("ERROR on accept");
```

Edge程式說明-2

2. 得到Edge UE的畫面

```
cv::VideoCapture vcap,vcap2;
int count=1;
int count_t = 1;
const std::string videoStreamAddress = "rtmp://163.18.104.132:1935/myapp/test"; //Edge UE frame (background)
// const std::string videoStreamAddress2 = "cloud.mp4";
// const std::string videoStreamAddress2 = "camera1.mp4";

Mat background;

if(!vcap.open(videoStreamAddress)) {
    std::cout << "Error opening video stream or file" << std::endl;
    return -1;
}
```

Edge程式說明-3

3. 接收與還原由Cloud Server擷取後的物件

```
uchar sock[3];

Mat img = Mat::zeros(FRAME_HEIGHT, FRAME_WIDTH, CV_8UC3);
imgSize = img.total()*img.elemSize();
uchar sockData[imgSize];

for(int i=0;i<imgSize;i+=bytes)
    if ((bytes=recv(newsockfd, sockData+i, imgSize-i,0))==-1) error("recv failed");

Mat object ;

object = Mat::zeros(FRAME_HEIGHT, FRAME_WIDTH, CV_8UC3);
int ptr=0;
int i,j;
for(i=0;i<img.rows;++i)
    for(j=0;j<img.cols;++j)
    {
        object.at<Vec3b>(i,j) = Vec3b(sockData[ptr+0],sockData[ptr+1],sockData[ptr+2]);
        ptr=ptr+3;
    }
imshow( "object", object );
```

Edge程式說明-4

4. 對物件與Edge UE的畫面做重疊

```
Mat overlay;

overlay = Mat::zeros(FRAME_HEIGHT, FRAME_WIDTH, CV_8UC3);
ptr =0;
for(i=0;i<overlay.rows;++i)
for(j=0;j<overlay.cols;++j)
{
    if((object.at<Vec3b>(i,j)[0]!=0)|| (object.at<Vec3b>(i,j)[1]!=0)|| (object.at<Vec3b>(i,j)[2]!=0))
        overlay.at<Vec3b>(i,j) = Vec3b(object.at<Vec3b>(i,j)[0],object.at<Vec3b>(i,j)[1],object.at<Vec3b>(i,j)[2]); //pixel to pixel
    else
        overlay.at<Vec3b>(i,j) = Vec3b(background.at<Vec3b>(i,j)[0],background.at<Vec3b>(i,j)[1],background.at<Vec3b>(i,j)[2]);
    //printf("%d %d %d\n",object.at<Vec3b>(i,j)[0],object.at<Vec3b>(i,j)[1],object.at<Vec3b>(i,j)[2]);
    ptr=ptr+3;
}
imshow( "overlay", overlay );
imwrite("overlay.jpg", overlay);
```

Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

實驗流程

• 實驗的五個步驟：

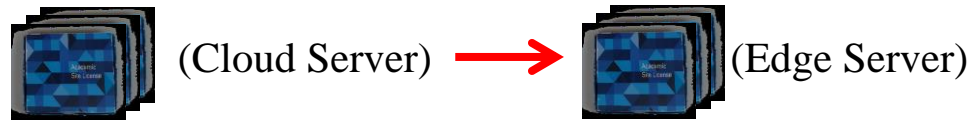
一. Cloud UE將畫面傳送到Cloud Server， Edge UE將畫面傳送到Edge Server



二. Cloud Server將Cloud UE的畫面做擷取



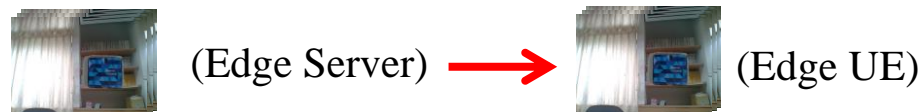
三. Cloud Server傳送(socket)擷取後的物件到Edge Server



四. Edge Server將物件與Edge UE的畫面做重疊



五. Edge UE播放重疊後的畫面



OAI運行(啟動EPC程式)

在終端機輸入

- `cd ~/openair-cn/scripts`
- `./run_hss`

開啟一個新的終端機，並且輸入

- `cd ~/openair-cn/scripts`
- `./run_mme`

開啟一個新的終端機，並且輸入

- `cd ~/openair-cn/scripts`
- `./run_spgw`

OAI運行(啟動eNB程式)

到eNB的電腦上，在終端機輸入

- `$ cd ~/openairinterface5g/cmake_targets/lte_build_oai/build`
- `$ sudo -E ./lte-softmodem -O ~/openairinterface5g/targets/PROJECTS/GENERIC-LTE-EPC/CONF/enb.band7.tm1.usrpb210.conf -d`

UE註冊

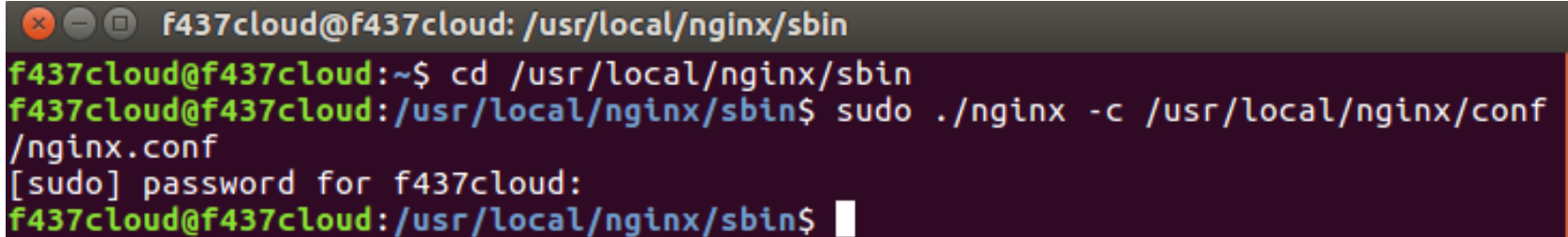
到網頁上輸入192.168.8.1，並開啟LTE網路



開啟 Video Streaming Server-1

開啟 Cloud Server的 Video Streaming Server

- `$ cd /usr/local/nginx/sbin`
- `$ sudo ./nginx -c /usr/local/nginx/conf/nginx.conf`

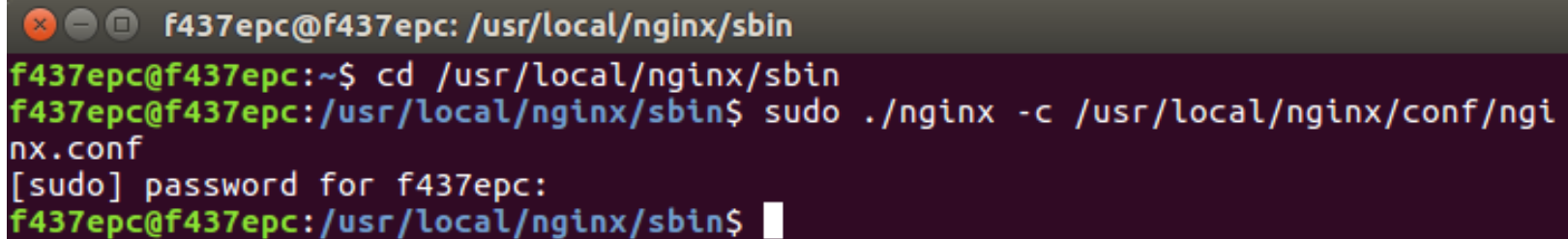


```
f437cloud@f437cloud: /usr/local/nginx/sbin
f437cloud@f437cloud:~$ cd /usr/local/nginx/sbin
f437cloud@f437cloud:/usr/local/nginx/sbin$ sudo ./nginx -c /usr/local/nginx/conf/nginx.conf
[sudo] password for f437cloud:
f437cloud@f437cloud:/usr/local/nginx/sbin$
```

開啟 Video Streaming Server-2

開啟 Edge Server的 Video Streaming Server

- `$ cd /usr/local/nginx/sbin`
- `$ sudo ./nginx -c /usr/local/nginx/conf/nginx.conf`

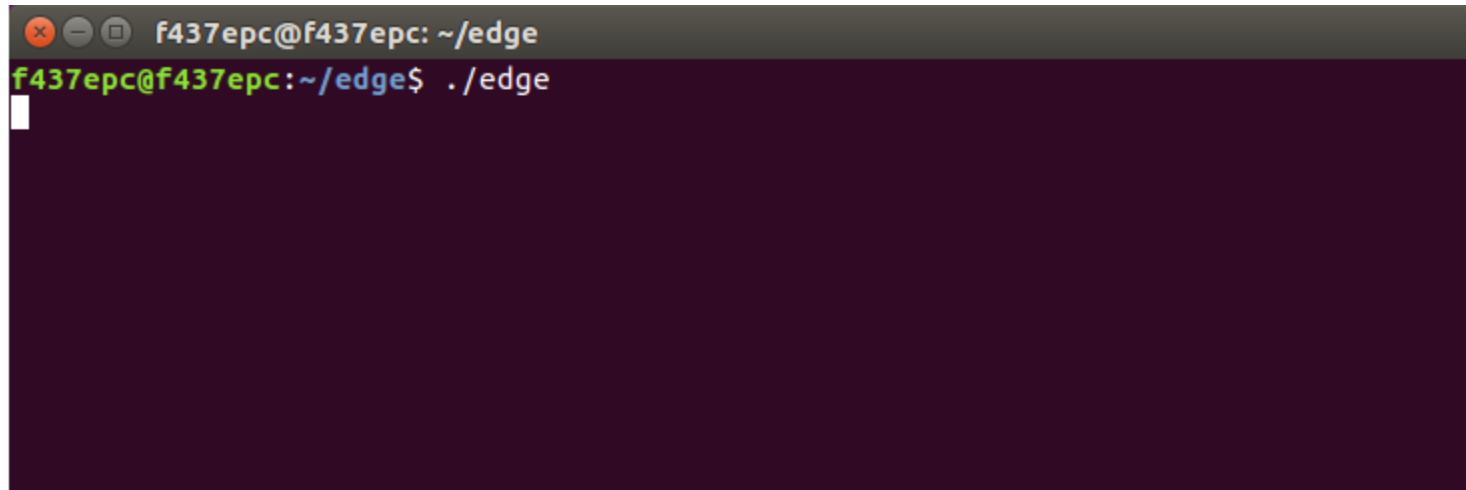


```
f437epc@f437epc: /usr/local/nginx/sbin
f437epc@f437epc:~$ cd /usr/local/nginx/sbin
f437epc@f437epc:/usr/local/nginx/sbin$ sudo ./nginx -c /usr/local/nginx/conf/nginx.conf
[sudo] password for f437epc:
f437epc@f437epc:/usr/local/nginx/sbin$
```

執行程式-Edge Server

執行Edge程式

- \$ cd ~/edge
- \$./edge

A terminal window with a dark purple background. The title bar shows window control icons and the text 'f437epc@f437epc: ~/edge'. The prompt 'f437epc@f437epc:~/edge\$' is followed by the command './edge' which has been executed. A white cursor is visible on the line below the command.

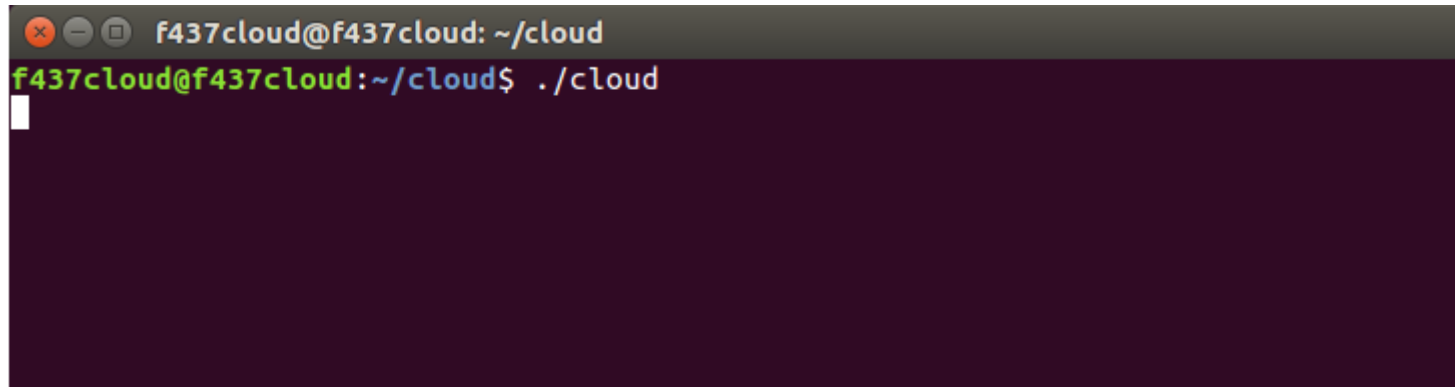
```
f437epc@f437epc: ~/edge
f437epc@f437epc:~/edge$ ./edge

```

執行程式-Cloud Server

執行Cloud程式

- \$ cd ~/cloud
- \$./cloud



```
f437cloud@f437cloud: ~/cloud
f437cloud@f437cloud:~/cloud$ ./cloud
```

UE_c上傳影片

UE_c上傳影片至Cloud Server

Cloud Server的IP

- `$ ffmpeg -re -i /dev/video0 -r 10 -q:v 15 -f flv rtmp://163.18.104.142:1935/myapp/test`

1. 設定FPS：`-r 30` (FPS=30)
2. 設定畫質：`-q:v 1` (1表示最好，30最差)
3. 設定格式：`-f flv`
4. 設定輸出位置：`rtmp://163.18.104.132:1935/myapp/test`

```
f437@f437: ~  
f437@f437:~$ ffmpeg -re -i /dev/video0 -r 10 -q:v 10 -f flv rtmp://163.18.104.14  
2:1935/myapp/test  
ffmpeg version 2.8.17-0ubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers  
built with gcc 5.4.0 (Ubuntu 5.4.0-6ubuntu1~16.04.12) 20160609  
configuration: --prefix=/usr --extra-version=0ubuntu0.1 --build-suffix=-ffmpeg  
--toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x  
86_64-linux-gnu --cc=cc --cxx=g++ --enable-gpl --enable-shared --disable-strippi  
ng --disable-decoder=libopenjpeg --disable-decoder=libschroedinger --enable-avre  
sample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enabl  
e-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-libflite  
--enable-libfontconfig --enable-libfreetype --enable-libfribidi --enable-libgme  
enable-libgsm --enable-libmodplug --enable-libmp3lame --enable-libopenjpeg
```

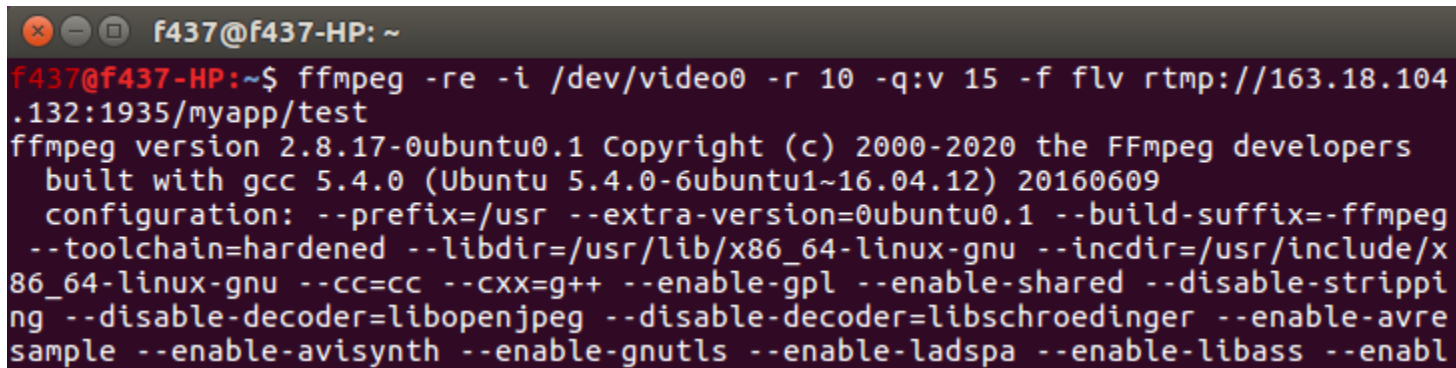

UE_e上傳影片

UE_e上傳影片 至Edge Server

Edge Server的IP

- `$ ffmpeg -re -i /dev/video0 -r 10 -q:v 15 -f flv rtmp://163.18.104.132:1935/myapp/test`

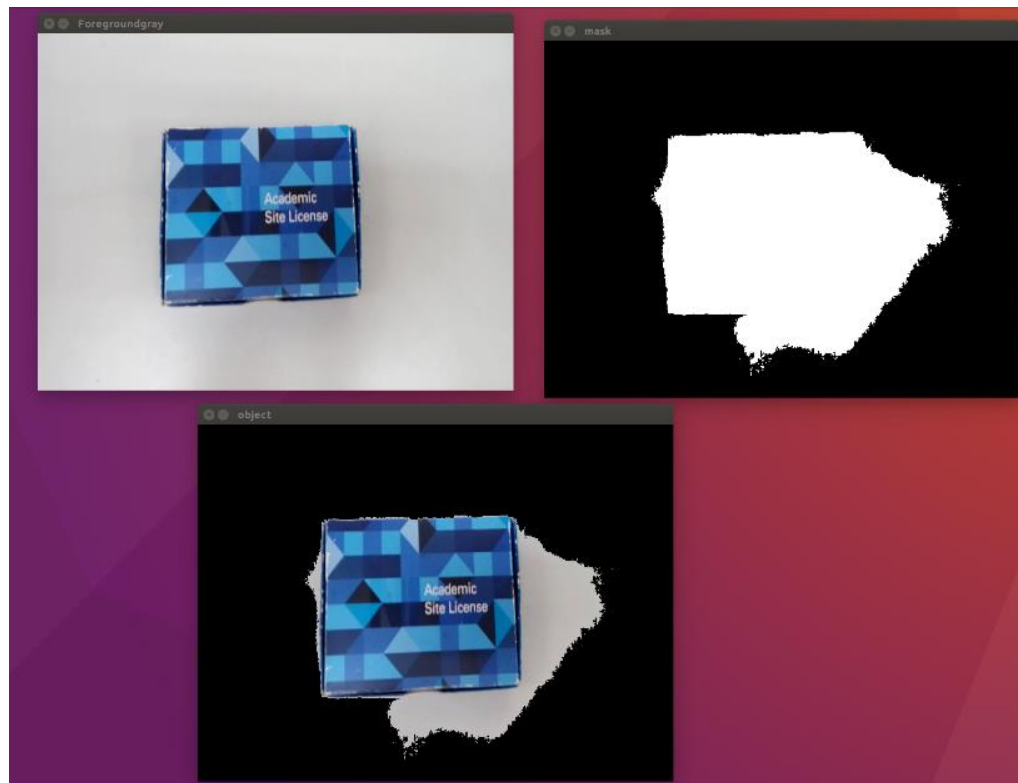
1. 設定FPS：`-r 30` (FPS=30)
2. 設定畫質：`-q:v 1` (1表示最好，30最差)
3. 設定格式：`-f flv`
4. 設定輸出位置：`rtmp://163.18.104.132:1935/myapp/test`



```
f437@f437-HP: ~  
f437@f437-HP:~$ ffmpeg -re -i /dev/video0 -r 10 -q:v 15 -f flv rtmp://163.18.104.132:1935/myapp/test  
ffmpeg version 2.8.17-0ubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers  
built with gcc 5.4.0 (Ubuntu 5.4.0-6ubuntu1~16.04.12) 20160609  
configuration: --prefix=/usr --extra-version=0ubuntu0.1 --build-suffix=-ffmpeg  
--toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x86_64-linux-gnu --cc=cc --cxx=g++ --enable-gpl --enable-shared --disable-stripping --disable-decoder=libopenjpeg --disable-decoder=libschroedinger --enable-avresample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enable-lib
```

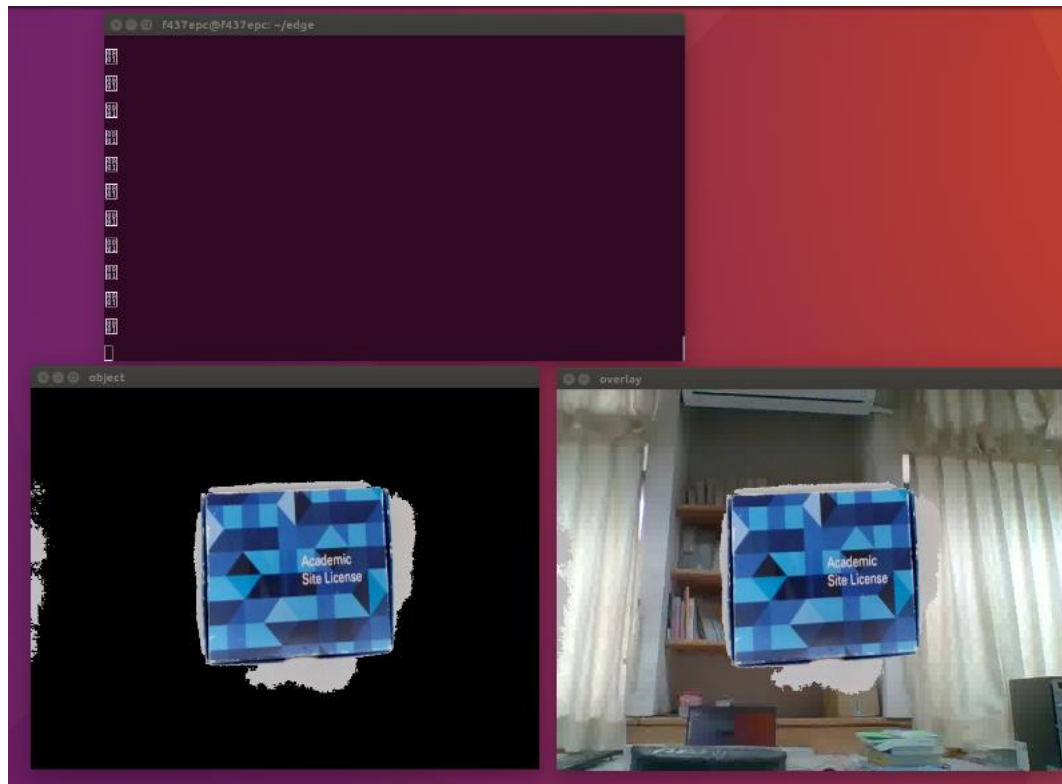
Cloud Server 畫面

- UE_c 和 UE_e 上傳影片後，Cloud Server呈現畫面



Edge Server 畫面

- UE_c 和 UE_e 上傳影片後，Edge Server呈現畫面



Edge Server上傳重疊影片

Edge Server上傳重疊後的影片

- `$ ffmpeg -f image2pipe -loop 1 -re -i "overlay.jpg" -r 10 -q:v 15 -f flv rtmp://163.18.104.132:1935/myapp/overlay`
Edge Server的IP

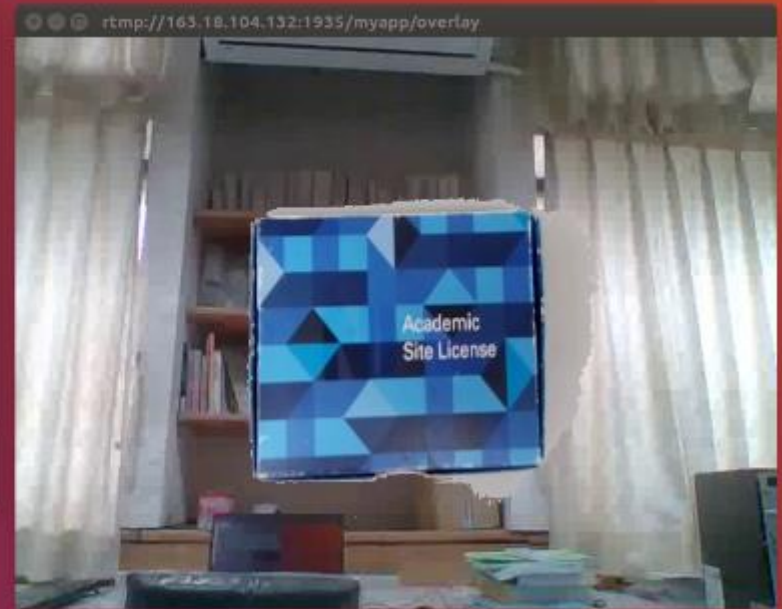
```
f437epc@f437epc: ~/edge
f437epc@f437epc:~/edge$ ffmpeg -f image2pipe -loop 1 -re -i "overlay.jpg" -r 30
-q:v 1 -f flv rtmp://163.18.104.132:1935/myapp/overlay
ffmpeg version 2.8.17-0ubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers
built with gcc 5.4.0 (Ubuntu 5.4.0-6ubuntu1~16.04.12) 20160609
configuration: --prefix=/usr --extra-version=0ubuntu0.1 --build-suffix=-ffmpeg
--toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/x
86_64-linux-gnu --cc=cc --cxx=g++ --enable-gpl --enable-shared --disable-strippi
ng --disable-decoder=libopenjpeg --disable-decoder=libschrödinger --enable-avre
sample --enable-avisynth --enable-gnutls --enable-ladspa --enable-libass --enabl
```

UE_e觀看重疊影片

UE_e觀看重疊影片

- \$ ffplay rtmp://163.18.104.132:1935/myapp/overlay

```
F437@F437-HP: ~  
displayWidth      640.00  
displayHeight     480.00  
duration          0.00  
framerate         30.00  
fps              30.00  
videodatarate     195.00  
videocodecid      2.00  
audiodatarate     0.00  
audiocodecid      0.00  
[live_flv @ 0x7f5ef0009280] Could not find codec parameters for stream 1 (Audio:  
none, 0 channels): unknown codec  
Consider increasing the value for the 'analyzeduration' and 'probesize' options  
Input #0, live_flv, from 'rtmp://163.18.104.132:1935/myapp/overlay':  
Metadata:  
  Server      : NGINX RTMP (github.com/arut/nginx-rtmp-module)  
  displayWidth : 640  
  displayHeight : 480  
  fps         : 30  
  profile     :  
  level      :  
Duration: 00:00:00.00, start: 0.000000, bitrate: N/A  
Stream #0:0: Video: flv1, yuv420p, 640x480, 199 kb/s, 30 tbr, 1k tbn, 1k tbc  
Stream #0:1: Audio: none, 0 channels  
[ 37.00 M-V: 0.000 fd= 0 aq= 0KB vq= 99KB sq= 0B f=0/0
```



Outline

- 實驗目的及實驗內容
- 實驗環境
- 平台安裝需求
- Edge與Cloud程式碼修改
- 實驗流程說明
- AR / VR 邊緣計算實驗執行
- 附錄

Edge Server 與 Cloud Server的程式碼

- Edge Server 與 Cloud Server的程式碼

下載網址：

<https://drive.google.com/drive/folders/1uljnvdo9FkSSTqVSuCYH45y3OyDDl4qH?usp=sharing>