

TRACING THE WAY OF DATA IN A TCP CONNECTION THROUGH THE LINUX KERNEL

R.Karthikeyan¹, Muthumani M², Muthuraj E³

¹ Asst.Prof, Dept of MCA, Gnanamani college of Technology, Namakkal, INDIA.

² P.G.Scholar, Dept of MCA, Gnanamani college of Technology, Namakkal, INDIA.

³ P.G.Scholar, Dept of MCA, Gnanamani college of Technology, Namakkal, INDIA.

Abstract--- *With the processing of CMOS technology, the technology of video acquisition based on CMOS is becoming a new trend. However, many CMOS camera chip is not supported by the newest Linux kernel yet. The environment image acquirement and the bluetooth transmission system are studied and designed . In this paper, the method of designing the CMOS camera driver based on S3C2440 developing board with the embedded Linux environment is introduced and add some components such as a usb bluetooth adapter. SCCB is a distinguishing feature of OV series CMOS chips. S3C2440 provides a camera interface, and the camera driver is designed based on it. The library and the utils are compiled, and of images got from CMOS camera to the Bluetooth mobile phone has been realized by means of programming .*

Keywords-- camera device driver; OV9650; SCCB; S3C2440; image acquirement; Bluetooth technique

1. Introduction

Comparing with traditional CCD image sensor, CMOS image sensors using CMOS technology can integrate the pixel array and peripheral circuits (such as the image sensor core, single-clock, all the sequential logic, programmable functions and A / D converter) in a chip, with a small size, light weight, low power consumption, programming convenience, easy to control and so on. Embedded Linux is a better embedded operation system, which has portable, strong network function and excellent GNU compile tools as well as free open source characters. S3C2410 is a 32 bits embedded processor that based on an ARM920T core with integrated MMU and abundant internal resources.

Because of low cost, high performance and the Video4linux (short for V4L) support on Linux program, CMOS camera is easy to integrate to embedded system. Bluetooth

technique is a short distance wireless communication with dominant position which is exclusive certificated, and it has been widely used in mobile phones, personal computer and peripherals at present. This paper introduce image acquirement function based on embedded Linux system, and transmit the picture got from CMOS camera to the Bluetooth mobile phone on the environment of Bluetooth that has been constructed.

2. Software & Hardware Platform Guide

The system uses SAMSUNG S3C2440 as a microprocessor, the camera equipment is OV9650. SAMSUNG S3C2440 uses 16/32 bit ARM920T RISC technology for the core. Its main Frequency is 400M Hz. It provides a camera interface (camif) to support camera. There are two models for camif to transmit data with DMA controller: one is called Preview mode, which transform the image data sampling from the camera interface into the RGB format, and transfer it to the SDRAM under control of the DMA; the other is called code mode, which transmits the image data to the SDRAM in YCbCr4:2:0 or 4:2:2 format. Software platform is embedded Linux OS.

We add a bank of NAND Flash storage of 64MB and two bands of SDRAM with 64KM total capability, and extend a ethernet network interface of 10MB by adding a ethernet network controller chip CS8900. There are two usb hosts, one is for CMOS camera, the other is for usb adapter. We acquire image data by CMOS camera and send the image data to the buffer, then save to file, and then transmit the file to Blue tooth mobile phone.

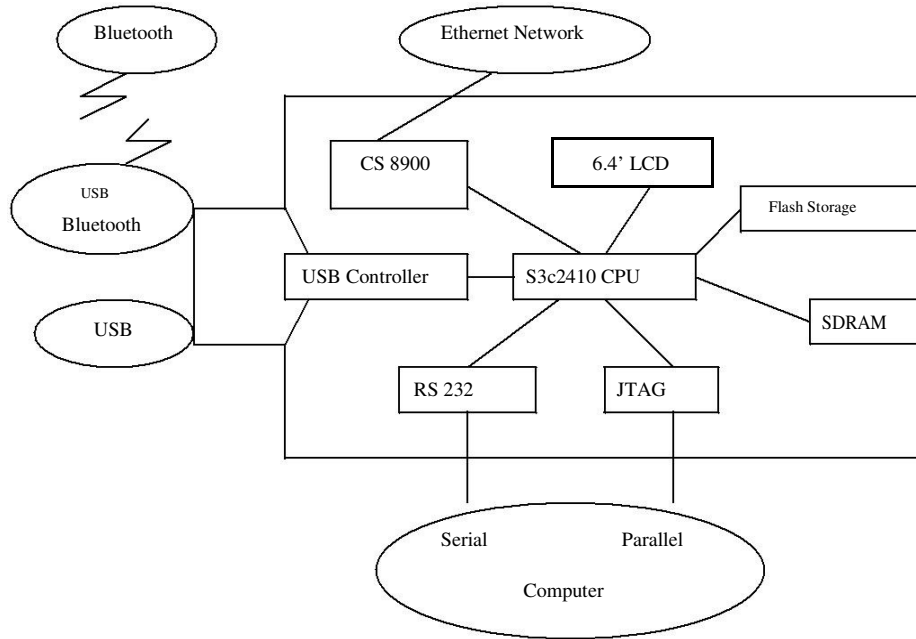


Fig 1. Hardware Platform

3. Implement of SCCB Driver

In Linux operating system, device driver provides access interfaces for underlying data structures, hardware device drivers and advanced data structures. Device driver shields the hardware details for the application. From Linux 2.6 kernel version, Linux operating system using kobject kernel object facility, which treat hardware device as a file. System calls and drivers are linked by file_operations. Every member of this structure corresponds to a system call. The S3C2440 does not provide sccb interface, so the driver has to include a sccb driver.

SCCB is short of serial camera control bus, which is customized by OminiVision Company. It is used for writing and reading the registers of the camera in order to control the output of the camera. SCCB data transmission is controlled by master device which send the data transmission start signal, the clock signals and stop signal. Usually the master device is microprocessor, and the sub device addressed access to it. Each device connected to the SCCB has a unique address (ID). At present, SCCB communication protocol supports only 100Kb/s or 400Kb/s transmission speed, and supports two kinds of address forms.

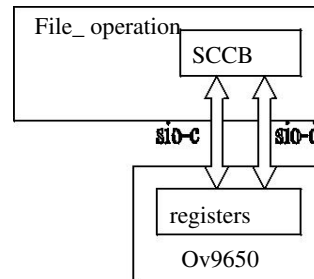


Fig 2 . Framework of the driver

Table1. Comparison of SCCB Read and Write Operation

Operation	Steps
three-phase write	Write the write address of the sub device (IDW,8bit) and the "Don't care"
	write the register address (8bit) and the "Don't care"
	write the data(8bit) and the "Don't care"
two-phase write	Write the write address of the sub device (IDW,8bit) and the "Don't care"
	writer the register address (8bit) and the "Don't care"
two-phase	Write the read address of the sub

Fig. 3 Algorithm flow chart to reading 8 bit

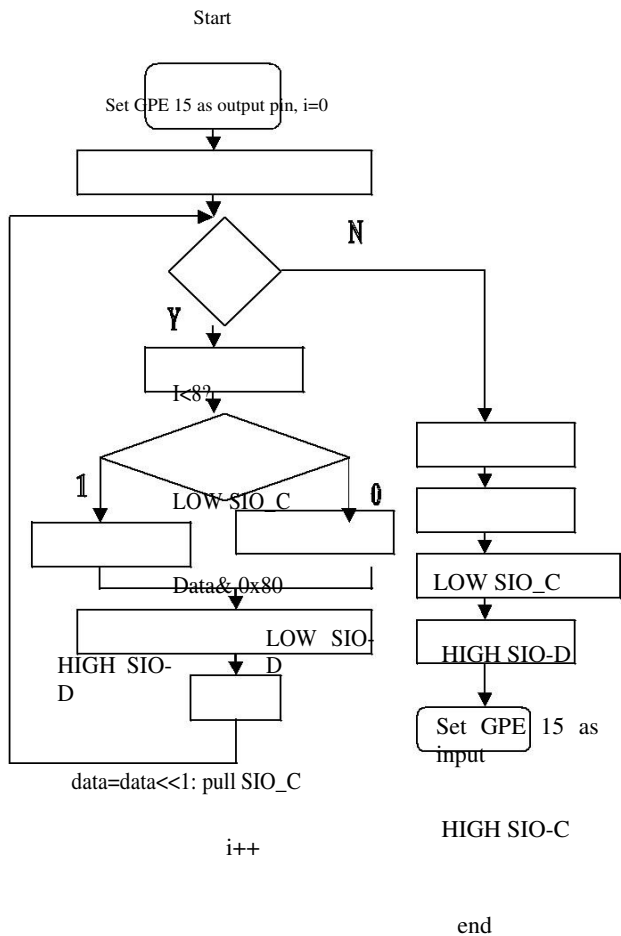
read the data in register and

Here, S3C2440 is the master device, OV9650 is the sub device. The important function: in SCCB driver are `sccb_write(u8 IdAddr, u8 SubAddr, u8 data)` which is used to write data into internal register of OV9650 and `u8 sccb_read(u8 IdAddr, u8 SubAddr)` which is used to read data from internal register of OV9650. Part of the code for `sccb_read(u8 IdAddr, u8 SubAddr)` is as follows

```
u8 sccb_read(u8 IdAddr, u8 SubAddr)
{
    u8 data;
    down(&bus_lock);
    sccb_start();
    sccb_write_byte(IdAddr);
    sccb_write_byte(SubAddr);
    sccb_stop();
```

```
    sccb_start();
    sccb_write_byte(IdAddr|0x01);
    data = sccb_read_byte();
    sccb_stop();
    up(&bus_lock);
    return data;
}
```

Function `sccb_read_byte()` and `sccb_write_byte()` is respectively to complete 8 bit read and write according to SCCB timing logic.



➤ **Registration and initialization**

Here register the camera as a miscdevice. All the miscdevice share one major device number: 10. Using `minor = MISC_DYNAMIC_MINOR` to get a sub device number. All the miscdevices form a list. Calling the function `misc_register()` to add the device to the list, and registration is completed. Initialization includes:

1) Using `s3c2410_gpio_cfgpin` to set GPIO to camera mode. Part code is as follows: `s3c2410_gpio_cfgpin(S3C2440_GPJ8, S3C2440_GPJ8_CAMPCLK);`

2) Using `request_mem_region()` to ask for memory resources for camera. Using `ioremap_nocache()` to map the memory resources to kernel space.

3) Initialize SCCB. Set `SIC_C` and `SIC_D` to high, which is the waiting status.

4) Initialize camera. Write the parameters (output format, AGC, sampling clock) into an array, and then using a cycle function to write parameters into the internal register of OV9650 through SCCB one by one.

➤ **Some important functions**

When an application calls the camera, the system uses the sub device number to find the miscdevice, and then call its `file_operations` struct, and finally operate the API. There are some important functions to provide: `open()`, `release()`, `read()`.

❖ **Open():**

a) Ask for continuous blank memory for P mode and C mode. To speed up data transfer, using a PingPong structure data buffer for data transfer, asking for 4 memory buffers for P mode and C mode respectively, and recycling these 4 buffers. When a frame data transmission is completed, that is, one buffer of the PingPong structure is full, interrupt handling program can capture the output data while the next frame data is transmitted.

b) Open the interrupt for P mode and C mode. Part of the code is as follows:

```
request_irq(IRQ_S3C2440_CAM_C,
on_camif_irq_c,
IRQF_DISABLED, "CAM_C",
pdev);
```

```
request_irq(IRQ_S3C2440_CAM_P, on_camif_irq_p,
IRQF_DISABLED, "CAM_P",
pdev);
```

❖ **Read():**

a) Close interrupt. Part of the code is:

```

disable_irq(IRQ_S3C2440_CAM_C);
disable_irq(IRQ_S3C2440_CAM_P);
    b) Call start_capture()

```

- c) Using copy_to_user() to transmit the data in driver buffer into user space.
- d) Open interrupt when capturing is end.

❖ **Release():**

- a) Close camera interface.
- b) Release interrupt.
- c) Release buffer memory.
- d) Call misc_deregister() to deregister the device.

4. V4L Introduction

Video4linux(short for V4L) is the API interface in Linux that is used to acquire video and audio data. Adding appropriate video acquirement device and corresponding driver to

it, we can realize the video/image acquirement, AF/FM broadcast and switch channel function. This can be widely use in remote meeting, video phone and video surveillance system. Every external device can be regard as special file in Linux, these files are called device file. We can read or write them as access ordinary files. Generally speaking, the camera device file that adopt V4L structure is "/dev/v4l/video0". V4L supports two methods to capture image: mmap and read. V4L defined some important data structure in "/usr /include/linux/videodev.h". In course of image acquirement, we can operate these data structure to get the image data. Open the video device Open("/dev/video0/ Initialize the CMOS camera (set dln,video_picture Set the attribute of the Image(alette,bppIn,width,

5. Image Acquirement

In course of image acquirement, we adopt the read method of V4L to read directly the device to get the image data of JPEG format in this paper.

A. Initializing Camera Device

In course of initialization, we appoint the camera device file "/dev/video0",call open() function to open the device file, initialize the member of vdIn data structure that we have defined ever, include device name(vd->videodevice), the width (vd->hdrwidth) and the height (vd->hdrheight) of the image, pixel(vd>bppIn), frame size (vd->framesizeIn) and allocate the storage room for frame data (vd->pFramebuffer).

B. Setting the Attribute of Image

The CMOS camera adopt in this paper supports JPEG image acquirement. Image palette is

to start capture under corresponding mode.

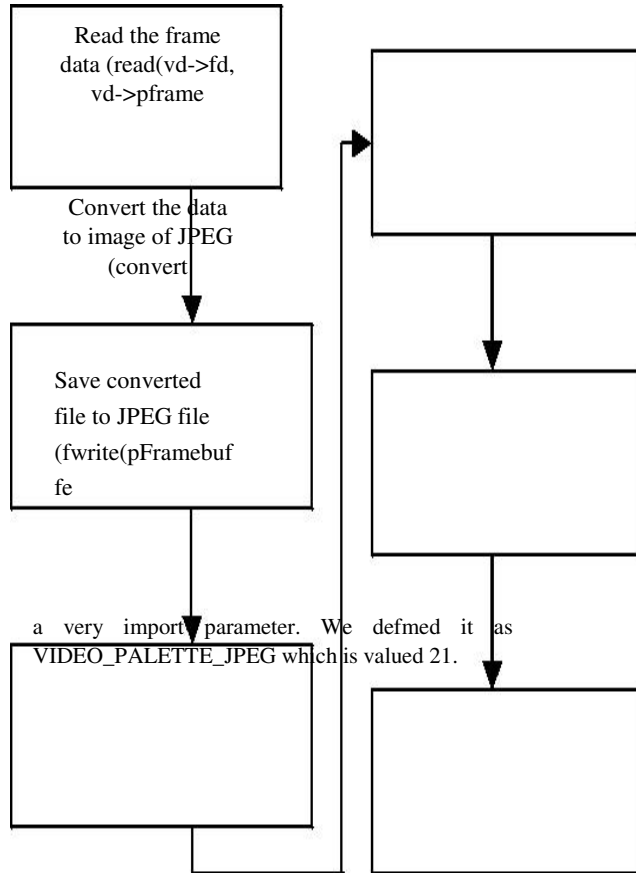
```

Printf("colour is % d\n", pict.colour) ;
Printf("contrast is %d\n", pict.contrast);
Printf("whiteness is % d\n", pict.
Whiteness);

```

C. Image Acquirement

In course of image acquirement, we adopt read method to read directly device file to get a frame data to save to the buffer, then call convertframe() function to convert the data in the pframebuffer to intact JPEG data and then save to ptframe buffer. Then we can call fwrite() function to save the JPEG data in ptframe buffer to appointed file read(vd->fd, vd->pFramebuffer, vd->framesizeIn); Jpegsize = convertframe(vd->ptframe, vd->pFramebuffer, vd->framesizeIn); fp=fopen("/tmp/ cap.jpeg","w"; fwrite(vd->ptframe, jpegsize,1, fp); fclose(fp);



Other parameter such as hue, colour, contrast can be getting from ioctl() that is called VIDIOCGPICT as its parameter. Ioctl (vd->fd, VIDIOCGPICT, &pict); printf("Brightness is %d\n", pict.brightness);

Fig.4 Flow of Image Acquirement

6. Bluetooth Transmission

Specification of the bluetooth protocol keeps to Open System Interconnection/Referenced Model (OSI/RM). OBEX is normally layered above RFCOMM in Bluetooth Protocol structure. OBEX supply presentation mode of the object, and transmit object by "Put" and "Get" operation. The session between devices follow the "request—answer" mode. The OBEX session begin send out connection request by client, server send out successful connection answer after connection has been built, otherwise server send out unsuccessful connection answer. After connection, client may push object to server by Put request, and if object is biggish, Put request may use many Put request

group, every request group need return a Put answer group. The acquired image in this paper has been saved to "/tmp/cap.jpg". Now we can transmit the picture to the Bluetooth mobile phone, the implement process as follow:

- **Patched the kernel and added the support of Bluetooth**
- **protocol, added usb bluetooth adapter driver.**
- **Cross compiled the bluez-lib and bluez-utils.**
- **Started usb adapter**
- **Realized the file transmission**

Conclusion

Nowadays, more and more image capture system uses CMOS camera. With the continuous progress of CMOS technology, CMOS image sensors are more and more high resolution, low-cost, small size and easy programming, its application will become increasingly widespread. In this paper, the method of designing the CMOS camera driver based on S3C2440 developing board and Image acquirement in embedded linux is realized. We constructed the working environment of bluetooth, and completed the file transmission between embedded Linux platform and bluetooth mobile phone by bluetooth OBEX protocol and Object PUSH

service. We realize remote surveillance by embedded system and wireless transmission.

References

1. El Gamal A.Eltoukhy H.CMOS image sensors[J].USA Circuits and Devices Magazine IEEE, 2005, 21(3) : 6-20
2. R.Karthikeyan," Improved Apriori Algorithm for Mining Rules" in the International Journal of Advanced Research in biology Engineering science and Technology Volume 11, Issue 4, April 2016, Page No:71-77.
3. R.Karthikeyan, & et all "Honeypots for Network Security", International journal for Research & Development in Technology. Volume 7. Issue 2 ,Jan 2017, Page No.:62-66 ISSN:2349-3585
4. Yuan Weiqi, Tang Yonghua. Application of SCCB of OmniVision camera in DSP[J]. Chinese Journal of Scientific Instrument. 2006_27_6__1687-1688. (in Chinese)
5. R.Karthikeyan,"A Survey on Position Based Routing in Mobile Adhoc Networks" in the international journal of P2P Network Trends and Technology, Volume 3 Issue 7 2013, ISSN:2249-2615, Page No.:81-88.
6. R.Karthikeyan,"A Survey on Sensor Networks" in the International Journal for Research & Development in Technology Volume 7, Issue 1, Jan 2017, Page No:71-77.
7. R.Karthikeyan, & et all "Web Based Honeypots Network", in the International journal for Research & Development in Technology. Volume 7. Issue 2 ,Jan 2017, Page No.:67-73 ISSN:2349-3585.

8. R.Karthikeyan, & et all, "A Simple Transmit Diversity Technique for Wireless Communication", in the International journal for Engineering and Techniques. Volume 3. Issue 1, Feb 2017, Page No.:56-61 ISSN:2395-1303.
9. R.Karthikeyan, & et all "Strategy of Tribble – E on Solving Trojan Defense in Cyber Crime Cases", International journal for Research & Development in Technology. Volume 7. Issue 1, Jan 2017, Page No.:167-171.
10. C. Ijeh1, Allan J. Brimicombe, David S. Preston Chris .O. Imafidon Security Measures in Wired and Wireless Networks Anthony , pg 113-12.
11. Karthikeyan, & et all "Advanced Honey Pot Architecture for Network Threats Quantification" in the international journal of Engineering and Techniques, Volume 3 Issue 2, March 2017, ISSN:2395-1303, PP No.:92-96.
12. DU Chun-lei. ARM Architecture and Programming[M]. Beijing _Tsinghua University Press, 2003. (in Chinese)
13. R.Karthikeyan, & et all "Estimating Driving Behavior by a smart phone" in the international journal of Engineering and Techniques, Volume 3 Issue 2, March 2017, ISSN:2395-1303, PP No.:84-91.
14. R.Karthikeyan, & et all "SAMI: Service- Based Arbitrated Multi-Tier Infrastructure for Cloud Computing" in the international journal for Research & Development in Technology, Volume 7 Issue 2, Jan 2017, ISSN(0):2349-3585, Pg.no:98-102
15. R.Karthikeyan, & et all "FLIP-OFDM for Optical Wireless Communications" in the international journal of Engineering and Techniques, Volume 3 Issue 1, Jan - Feb 2017, ISSN:2395-1303, PP No.:115-120.
16. R.Karthikeyan, & et all "Application Optimization in Mobile Cloud Computing" in the international journal of Engineering and Techniques, Volume 3 Issue 1, Jan - Feb 2017, ISSN:2395-1303, PP No.:121-125.
17. R.Karthikeyan, & et all "The Sybil Attack" in the international journal of Engineering and Techniques, Volume 3 Issue 3, May - Jun 2017, ISSN:2395-1303, PP No.:121-125.
18. R.Karthikeyan, & et all "Securing WMN Using Hybrid HoneyPot System" in the international journal of Engineering and Techniques, Volume 3 Issue 3, May - Jun 2017, ISSN:2395-1303, PP No.:121-125.
19. R.Karthikeyan, & et all "Automated Predictive big data analytics using Ontology based Semantics" in the international journal of Engineering and Techniques, Volume 3 Issue 3, May - Jun 2017, ISSN:2395-1303, PP No.:77-81.
20. R.Karthikeyan, & et all "A Survey of logical Models for OLAP databases" in the international journal of Engineering and Techniques, Volume 3 Issue 3, May - Jun 2017, ISSN:2395-1303, PP No.:171-181.
21. R.Karthikeyan, & et all "A Client Solution for Mitigating Cross Site Scripting Attacks" in the international journal of Engineering Science & Computing, Volume 7, Issue 6, June 2017, ISSN(0):2361-3361, PP No.:13063-13067.
22. R.Karthikeyan, & et all "A Condensation Based Approach to Privacy Preserving Data Mining" in the international journal of Engineering Science & Computing, Volume 7, Issue 6, June 2017, ISSN(0):2361-3361, PP No.:13185-13189.

23. R.Karthikeyan, & et all "Biometric for Mobile Security" in the international journal of Engineering Science & Computing, Volume7,Issue6, June 2017, ISSN(0):2361-3361,PP No.:13552-13555.
24. R.Karthikeyan, & et all "Data Mining on Parallel Database Systems" in the international journal of Engineering Science & Computing, Volume7,Issue7, July 2017, ISSN(0):2361-3361,PP No.:13922-13927.
25. R.Karthikeyan, & et all "Ant Colony System for Graph Coloring Problem" in the international journal of Engineering Science & Computing, Volume7,Issue7, July 2017, ISSN(0):2361-3361,PP No.:14120-14125.
26. R.Karthikeyan, & et all "Classification of Peer -To- Peer Architectures and Applications" in the international journal of Engineering Science & Computing, Volume7,Issue8, Aug 2017, ISSN(0):2361-3361,PP No.:14394-14397.
27. R.Karthikeyan, & et all "Mobile Banking Services" in the international journal of Engineering Science & Computing, Volume7,Issue7, July 2017, ISSN(0):2361-3361,PP No.:14357-14361
28. Zhang Da-bo. Principles, Design and Application of Embedded System[M].Beijing_Machinery Industry Press_2004. (in Chinese)
29. Samsung. S3C2440A 32Bit CMOS Microcontroller User's Manual[Z]. Samsung Electronics Corp, 2003.
30. R.Karthikeyan, & et all "Neural Networks for Shortest Path Computation and Routing in Computer Networks" in the international journal of Engineering and Techniques, Volume 3 Issue 4, Aug 2017, ISSN:2395-1303,PP No.:86-91.
31. R.Karthikeyan, & et all "An Sight into Virtual Techniques Private Networks & IP Tunneling" in the international journal of Engineering and Techniques, Volume 3 Issue 4, Aug 2017, ISSN:2395-1303,PP No.:129-133.
32. R.Karthikeyan, & et all "Routing Approaches in Mobile Ad-hoc Networks" in the International Journal of Research in Engineering Technology, Volume 2 Issue 5, Aug 2017, ISSN:2455-1341, Pg No.:1-7.
33. Jonathan C, Alessandro R, Greg K. Linux device drivers[M].3rd ed. Sebastopl, CA:O' Reilly & Associates, 2006:324-327.
34. R.Karthikeyan, & et all "Public Key Infrastructure Using Wireles Commnication Networks" in the International Journal of Computer Techniques, Volume 4 Issue 4, Aug 2017, ISSN:2394-2231, Pg No.:154-158.
35. R.Karthikeyan, & et all "Personal Identification based on Iris Texture Analysis" in the International Journal of Computer Techniques, Volume 4 Issue 6, Dec 2017, ISSN:2394-2231, Pg No.:95-101.

