1.

```
hanna@EURLTP-371:~$ cd modules/
hanna@EURLTP-371:~/modules$ ls
a.out  hello.c  mod2.c  mod21.c
hanna@EURLTP-371:~/modules$ mkdir mod6
hanna@EURLTP-371:~/modules$ cd mod6
hanna@EURLTP-371:~/modules/mod6$ ls file.txt
ls: cannot access 'file.txt': No such file or directory
hanna@EURLTP-371:~/modules/mod6$ vi file.txt
hanna@EURLTP-371:~/modules/mod6$ ls file.txt
file.txt
hanna@EURLTP-371:~/modules/mod6$
```

2.

Absolute path: /home/hanna/modules/mod6/file.txt

```
hanna@EURLTP-371:~/modules/mod6$ pwd
/home/hanna/modules/mod6/
hanna@EURLTP-371:~/modules/mod6$ ls
file.txt
```

Relative path: ../mod6/file.txt

```
hanna@EURLTP-371:~/modules/mod6$ realpath file.txt
/home/hanna/modules/mod6/file.txt
```

3.

```
hanna@EURLTP-371:~/modules/mod6$ vi file.txt
```

i) hanna@EURLTP-371:~/modules/mod6$ wc file.txt

```
  41  553 3739 file.txt
```

ii) hanna@EURLTP-371:~/modules/mod6$ head file.txt

```
Real time face detection has many challenges such as non-frontal faces, small scale
faces, occlusions, and complex backgrounds. Real time face detection can be done by using
Convolutional Neural Network (CNN) methods which produce high performance but have a
large amount of computation. It needs to be implemented on high computational device to
produce more accurate face detection results for any resolution image. To resolve the
challenges faced during real time face detection such as occlusions, different scales, different
backgrounds, varying positions, lighting and poses are detected accurately using proposed
framework. The amount of computation required for real time face detection is reduced with
low computational device (CPU) only. Real-time Face Detection is done efficiently by using
light architecture based on deep CNN. Three modules have been designed for our proposed
```

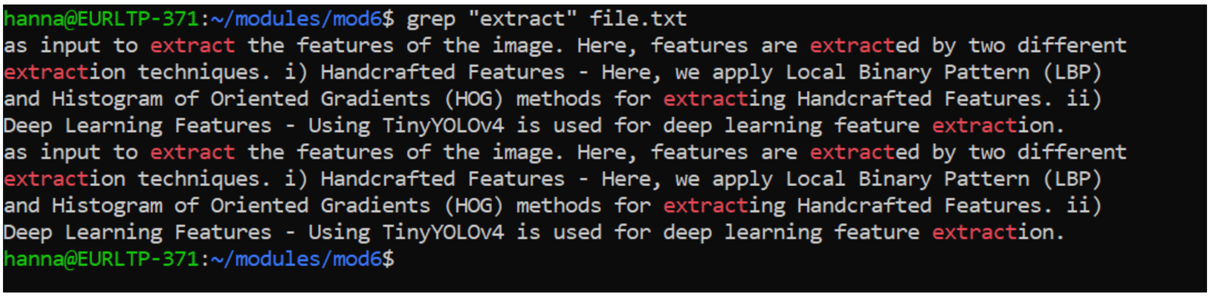
iii) hanna@EURLTP-371:~/modules/mod6$ cat file.txt

```
Real time face detection has many challenges such as non-frontal faces, small scale
faces, occlusions, and complex backgrounds. Real time face detection can be done by using
Convolutional Neural Network (CNN) methods which produce high performance but have a
large amount of computation. It needs to be implemented on high computational device to
produce more accurate face detection results for any resolution image. To resolve the
```

challenges faced during real time face detection such as occlusions, different scales, different backgrounds, varying positions, lighting and poses are detected accurately using proposed framework. The amount of computation required for real time face detection is reduced with low computational device (CPU) only. Real-time Face Detection is done efficiently by using light architecture based on deep CNN. Three modules have been designed for our proposed real time face detector such as 1) Image Pre-processing -In image pre-processing step, input image will be converted to Grayscale, then Resizing is applied and finally Horizontal and Vertical Shifting is performed. 2) Feature Extraction - In this step, pre-processed image is given as input to extract the features of the image. Here, features are extracted by two different extraction techniques. i) Handcrafted Features - Here, we apply Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG) methods for extracting Handcrafted Features. ii) Deep Learning Features - Using TinyYOLOv4 is used for deep learning feature extraction. After obtaining features from both techniques, hybrid feature fusion is applied. 3) Feature Selection - Here, important features are selected for reducing false positive rate. After this, classifier is applied to identify the presence of faces in the image and finally image with faces detected is displayed as outputReal time face detection has many challenges such as non-frontal faces, small scale

faces, occlusions, and complex backgrounds. Real time face detection can be done by using Convolutional Neural Network (CNN) methods which produce high performance but have a large amount of computation. It needs to be implemented on high computational device to produce more accurate face detection results for any resolution image. To resolve the challenges faced during real time face detection such as occlusions, different scales, different backgrounds, varying positions, lighting and poses are detected accurately using proposed framework. The amount of computation required for real time face detection is reduced with low computational device (CPU) only. Real-time Face Detection is done efficiently by using light architecture based on deep CNN. Three modules have been designed for our proposed real time face detector such as 1) Image Pre-processing -In image pre-processing step, input image will be converted to Grayscale, then Resizing is applied and finally Horizontal and Vertical Shifting is performed. 2) Feature Extraction - In this step, pre-processed image is given as input to extract the features of the image. Here, features are extracted by two different extraction techniques. i) Handcrafted Features - Here, we apply Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG) methods for extracting Handcrafted Features. ii) Deep Learning Features - Using TinyYOLOv4 is used for deep learning feature extraction. After obtaining features from both techniques, hybrid feature fusion is applied. 3) Feature Selection - Here, important features are selected for reducing false positive rate. After this, classifier is applied to identify the presence of faces in the image and finally image with faces detected is displayed as output

iv) Search a word 'X' in the entire file and display the lines with it



```
hanna@EURLTP-371:~/modules/mod6$ grep "extract" file.txt
as input to extract the features of the image. Here, features are extracted by two different
extraction techniques. i) Handcrafted Features - Here, we apply Local Binary Pattern (LBP)
and Histogram of Oriented Gradients (HOG) methods for extracting Handcrafted Features. ii)
Deep Learning Features - Using TinyYOLOv4 is used for deep learning feature extraction.
as input to extract the features of the image. Here, features are extracted by two different
extraction techniques. i) Handcrafted Features - Here, we apply Local Binary Pattern (LBP)
and Histogram of Oriented Gradients (HOG) methods for extracting Handcrafted Features. ii)
Deep Learning Features - Using TinyYOLOv4 is used for deep learning feature extraction.
hanna@EURLTP-371:~/modules/mod6$
```