

Computer Vision Assignment1 Report

학번 : 2019202032

이름 : 이상현

1. 과제 개요

1) RGB 4:4:4 컬러 포맷 영상을 YCbCr 4:4:4와 YCbCr 4:2:0 컬러 포맷 영상으로 변환하는 프로그래밍을 수행한다. 이때 입력영상은 raw file RGB image이며, opencv 함수를 사용하지 않고, YCbCr 4:4:4와 YCbCr 4:2:0 데이터로 변환하고 .yuv 파일로 저장한다. 또한 RGB 이미지와 변환된 YcbCr 이미지에 대해 각 채널 영상과 컬러 영상을 출력하도록 한다.

2) 30 frame의 4:2:0 YUV RGB 데이터를 입력으로 받아 RGB로 변환하여 opencv의 imshow() 함수를 사용하여 화면에 출력하고, 변환된 30 frame RGB 데이터를 .raw 파일로 저장할 수 있도록 한다.

2. 과제 수행 방법

1)

먼저 RGB 파일의 데이터를 fopen함수와 fread 함수를 사용하여 읽어온다. 이때 rgb 파일을 merge 함수를 사용하여 합칠 때는 b,g,r 순서로 저장이 되어있어야 원하는대로 파일이 저장된다는 것을 유의하여 rgb 파일의 정보를 받아와야 한다. 따라서 bgr[3]이라는 배열에 정보를 저장하고 merge한다면, bgr[0] 에 r, bgr[1]에 g, bgr[2]에 b를 저장한다. 이후 r, g, b와 merge한 데이터를 파일을 출력한다.

이후 r, g, b 데이터를 활용하여 y, cb, cr의 데이터로 변환한다. 데이터를 변환할 때는 다음의 식을 사용한다.

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cb = 128 - 0.169R - 0.331G + 0.500B$$

$$Cr = 128 + 0.500R - 0.419G - 0.0813B$$

352x288형태로 저장된 rgb데이터를 for문을 이용하여 각각의 요소에 접근하고, 위의 식을 사용하여 ycbcr 4:4:4 압축 데이터를 만든다.

YCbCr 4:2:0의 경우 Cb, Cr의 데이터가 Ycbcr 4:4:4에 비해 1/4만 저장되어 있으므로 위에서 생성한 352x288 형태의 Cb, Cr의 데이터를 resize하여 176x144 형태로 변환하고, merge 함수를 사용하여 YCbCr 4:2:0 형태의 파일을 만들고 저장한다.

위에서 만들어진 YCbCr4:4:4와 YCbCr 4:2:0 데이터를 yuv 파일로 저장하기 위해서 fwrite함수를 사용하여 파일로 작성한다. 이때 merge된 ycbcr데이터를 한번에 쓰는 것이 아니라 Y, Cb, Cr 데이터를 차례대로 한번씩 fwrite하여 파일을 완성한다.

2)

1번 과제에서와 마찬가지로 fopen과 fread 함수를 사용하여 데이터를 읽어온다. 이때 데이터는 Y, Cb, Cr 형태로 저장되어있고 Cb, Cr 데이터는 208x120형태로 저장되어 있으므로 resize 함수를 사용하여 416x240형태로 변환한다. Y, Cb, Cr 데이터를 아래와 같은 공식을 사용하여 RGB데이터로 변환한다.

$$R = 1.000Y + 1.402(Cr - 128)$$

$$G = 1.000Y - 0.714(Cr - 128) - 0.344(Cb - 128)$$

$$B = 1.000Y + 1.772(Cb - 128)$$

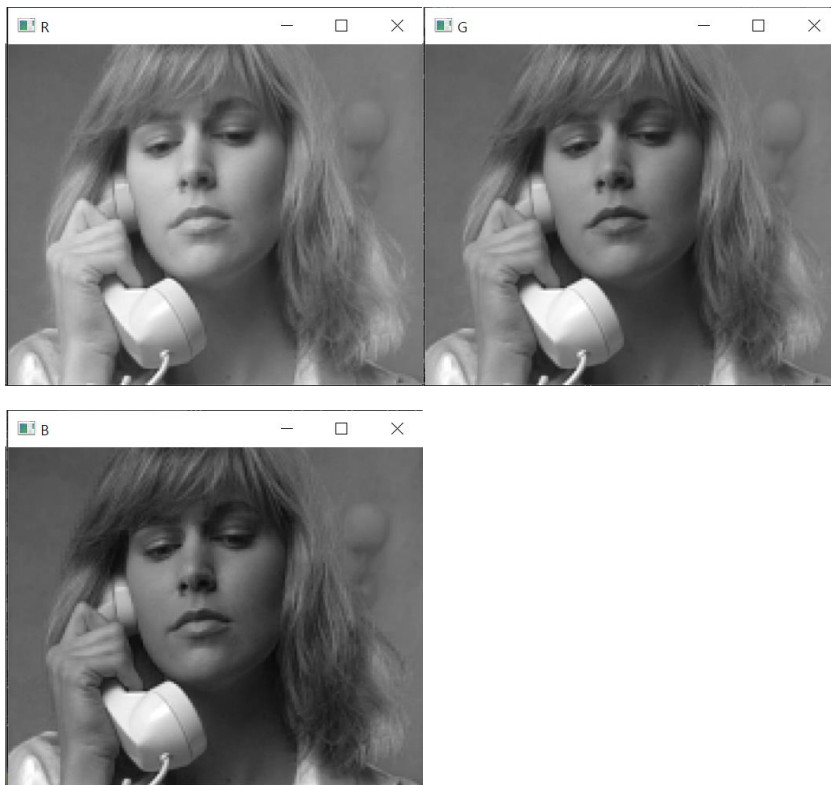
여기서 주의할 점은 RGB데이터를 merge할 때 b, g, r형태로 merge를 해야한다는 것이므로 변환된 r, g, b 데이터를 배열에 저장할 때는 0번지부터 2번지까지 b, g, r 순서대로 저장될 수 있도록 한다.

변환된 r, g, b 데이터를 1번에서와 같은 방법으로 fwrite 함수를 사용하여 raw 파일로 저장한다.

3. 결과 분석

1)

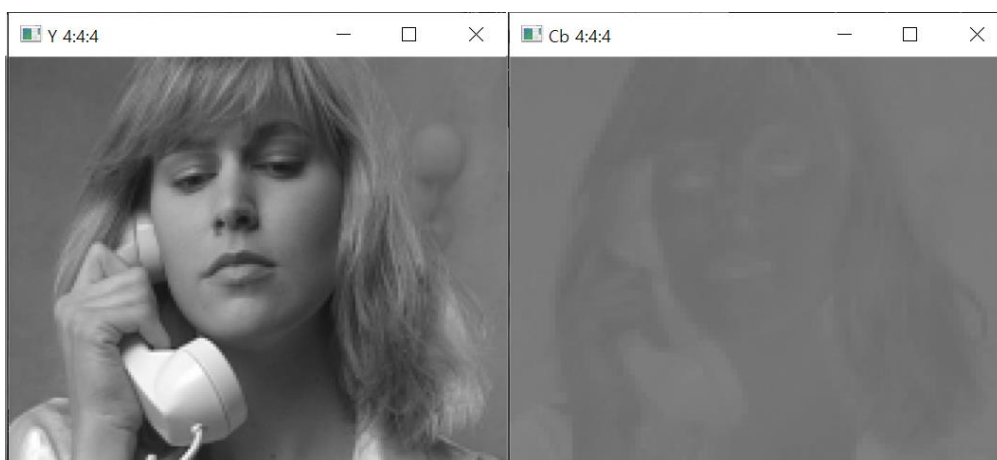
R, G, B 요소를 각각 출력하였고, R데이터의 경우 입술 쪽이 다른 데이터에 비해 상대적으로 밝다는 것을 통해 각각의 요소가 잘 출력되었음을 알 수 있다.

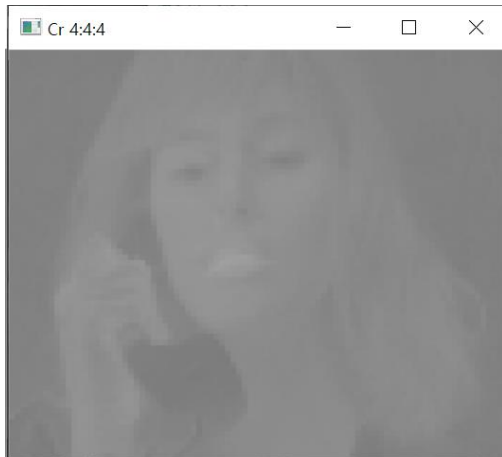


최종적으로 r, g, b 데이터를 merge하였다.

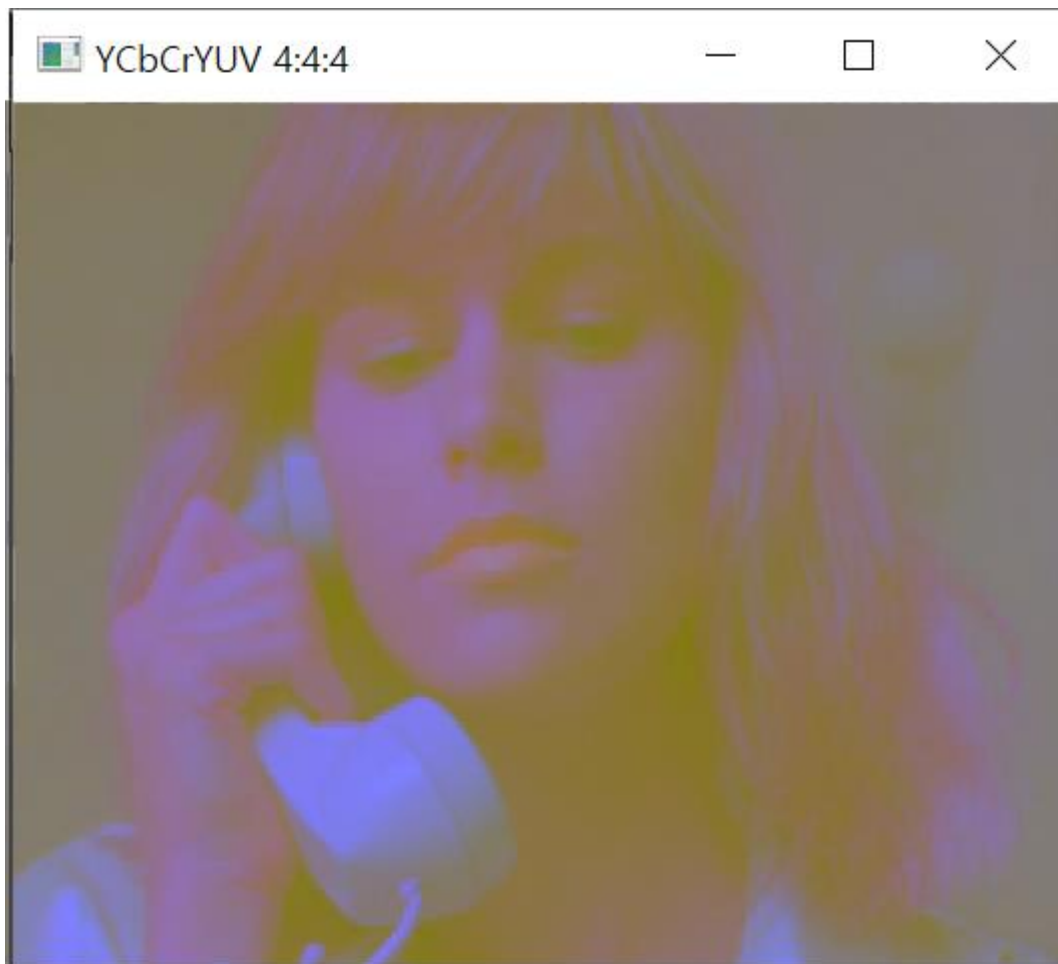


Y, cb,cr데이터를 각각 출력하였다. Cr의 데이터가 위의 r데이터에서 처럼 입술쪽이 상대적으로 밝게 나오는 것을 알 수 있다.





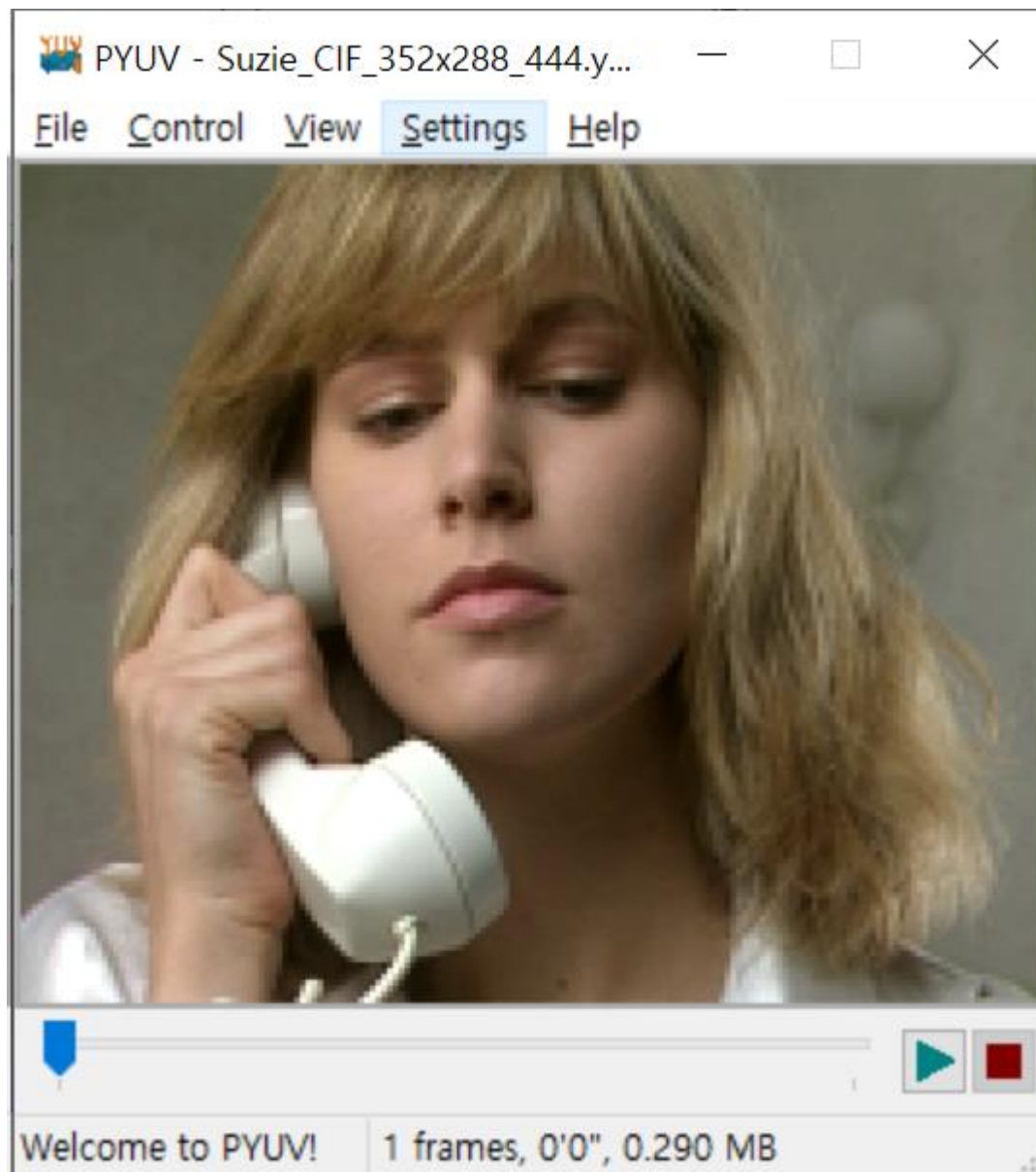
해당 데이터를 merge할 경우 다음과 같은 이미지가 나온다.



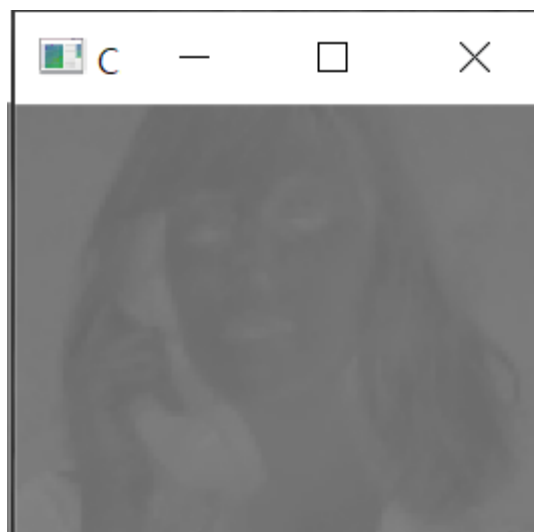
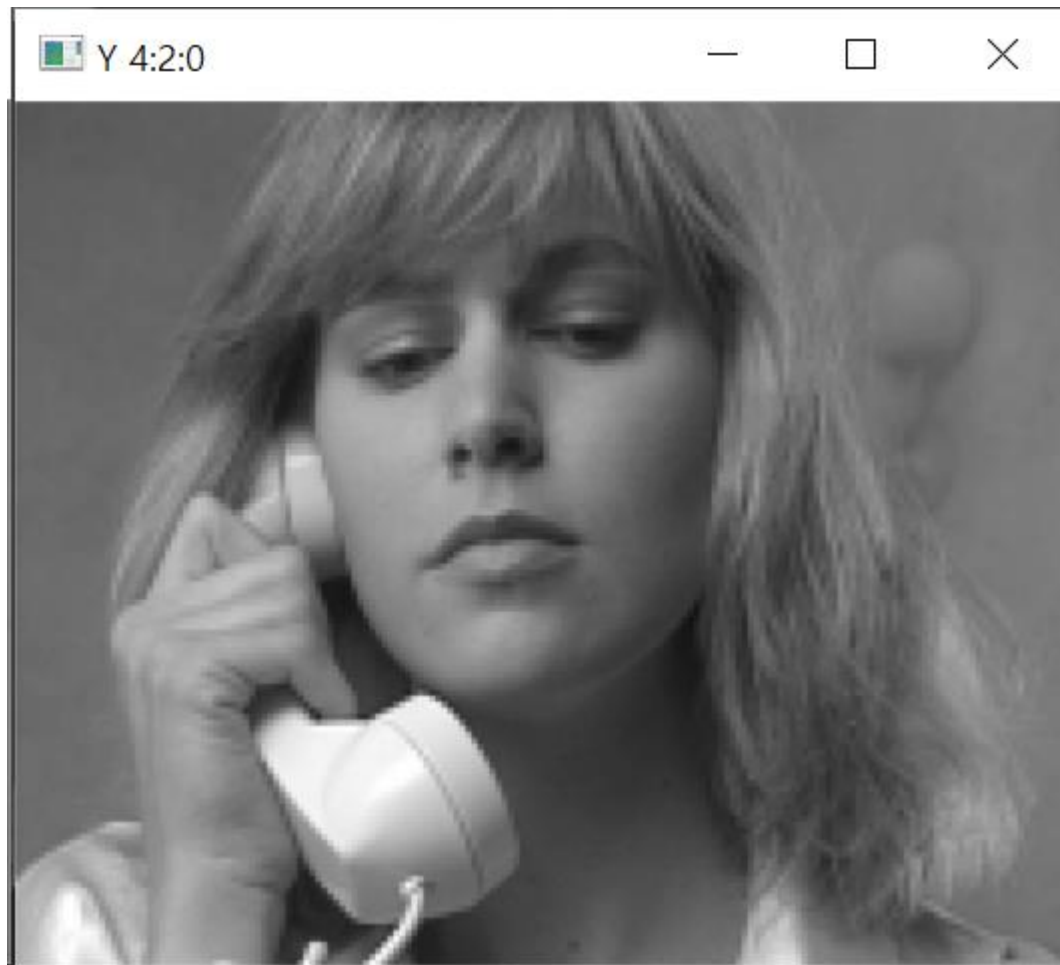
해당 데이터를 rgb데이터로 변환하고 merge하여 출력할 경우 다음과 같은 이미지가 나온다.

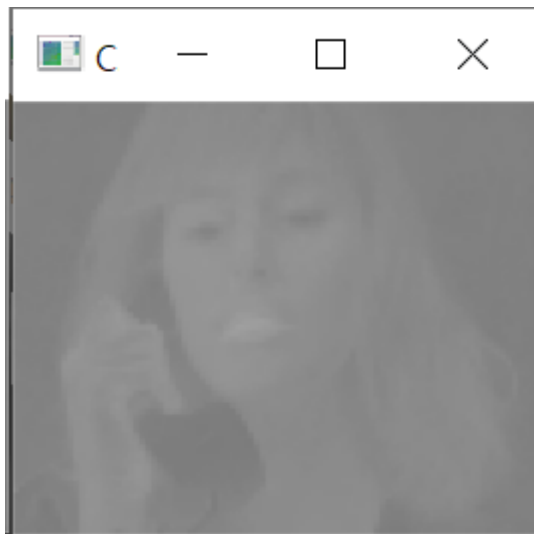


ycbcr데이터를 pyuv 파일로 열 경우 다음과 같은 결과를 확인할 수 있다.

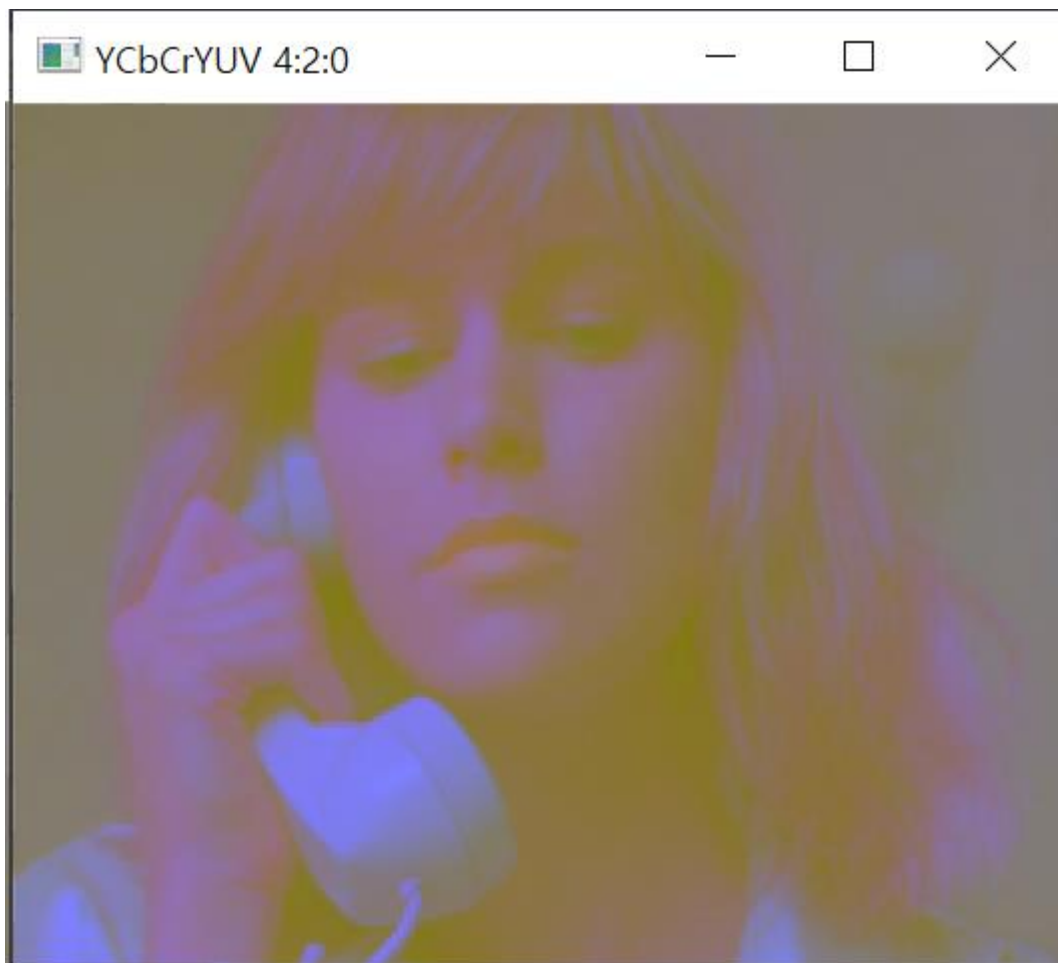


다음은 4:2:0으로 압축한 결과이다.

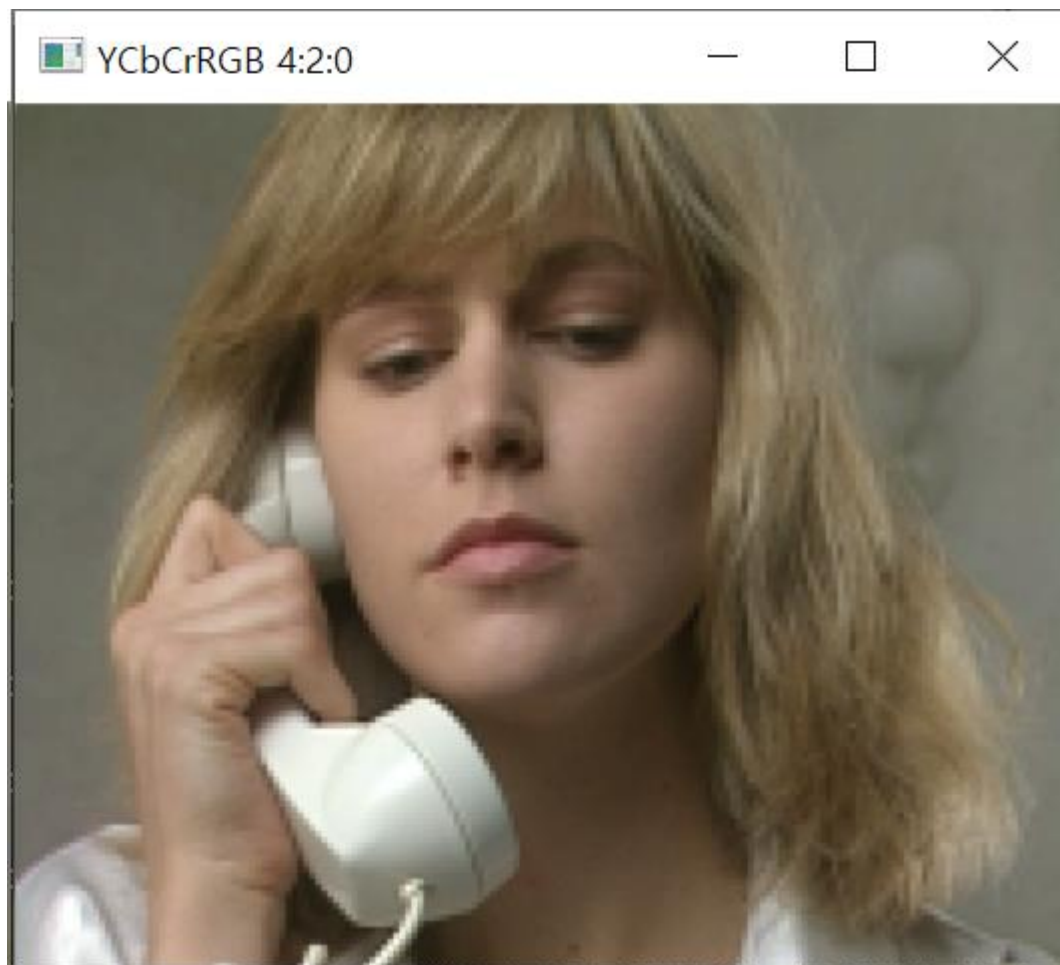




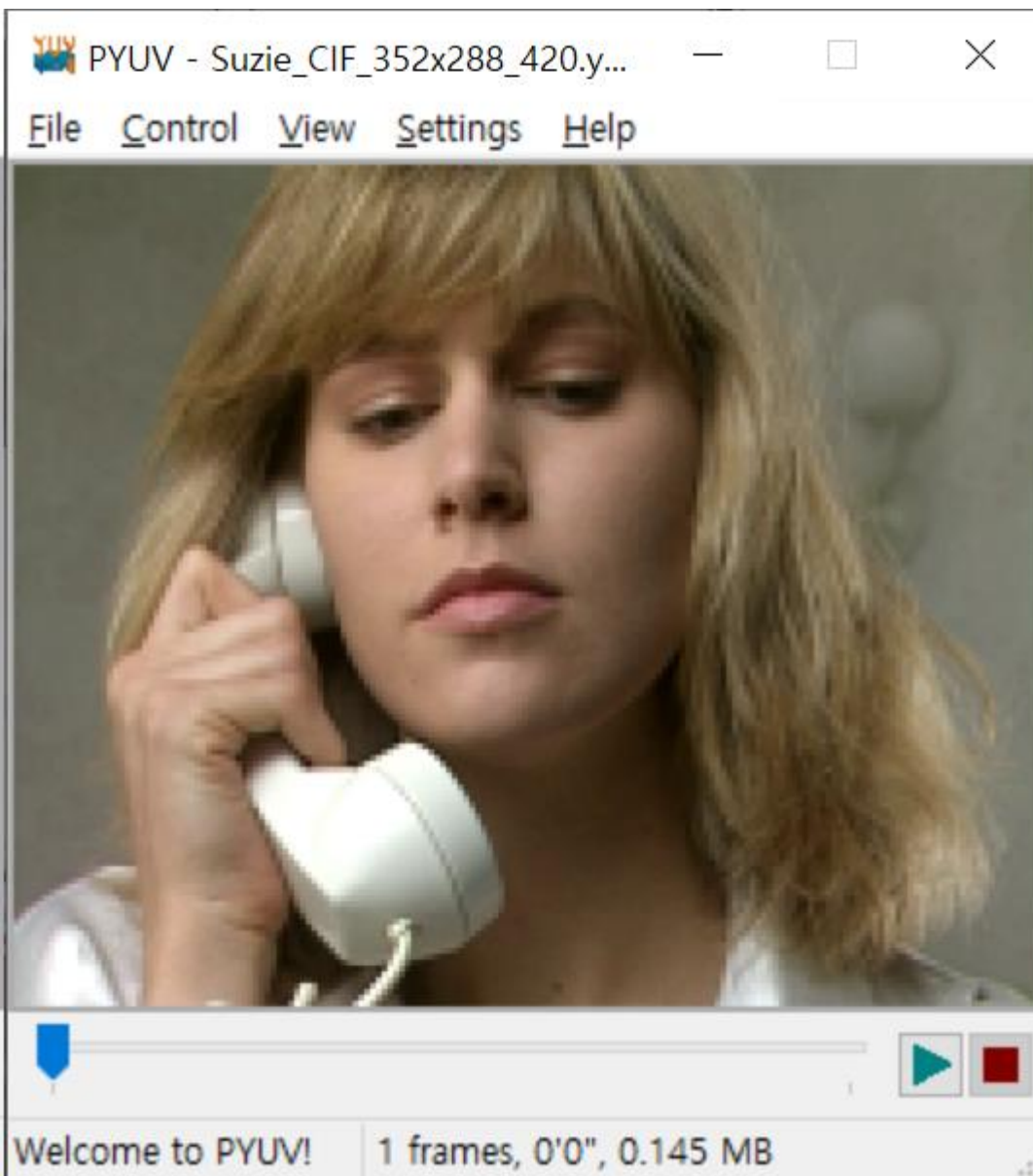
위의 데이터를 MERGE하면 다음과 같은 결과가 나온다.



위의 데이터를 RGB로 변환하여 저장하면 아래와 같은 결과가 나온다.

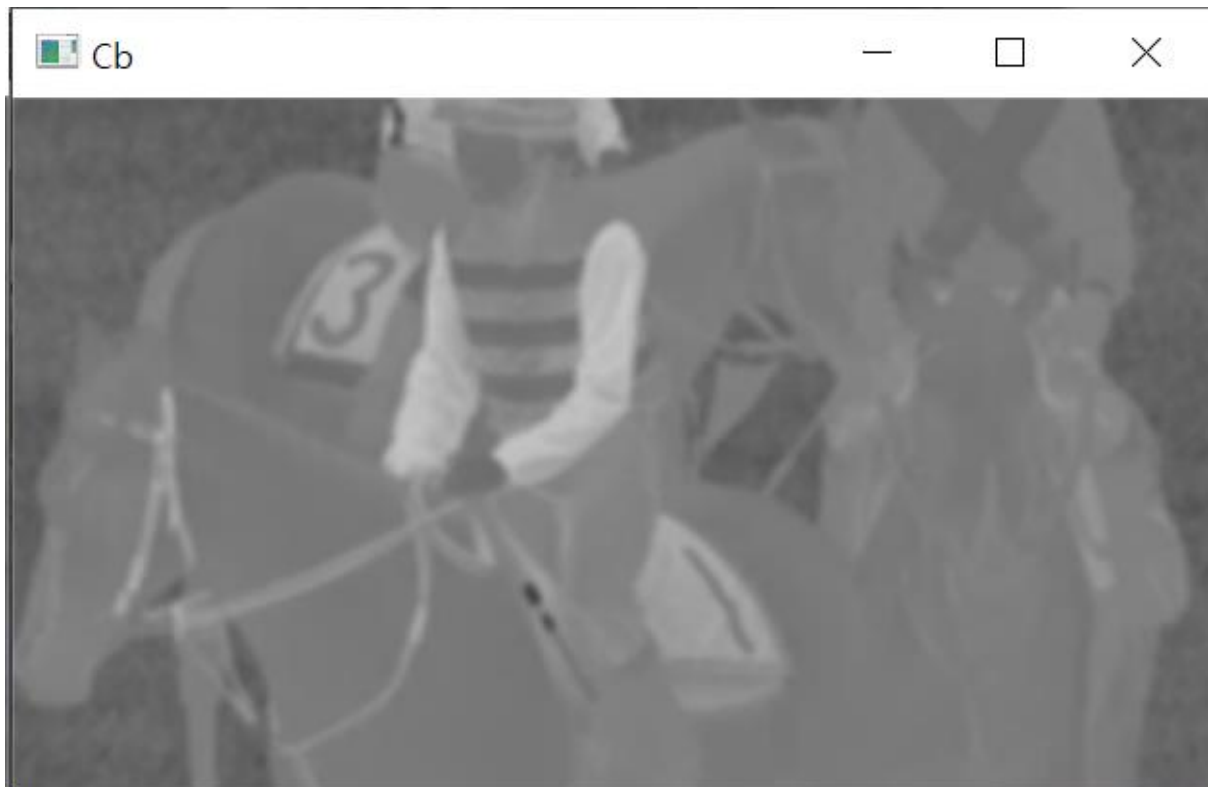
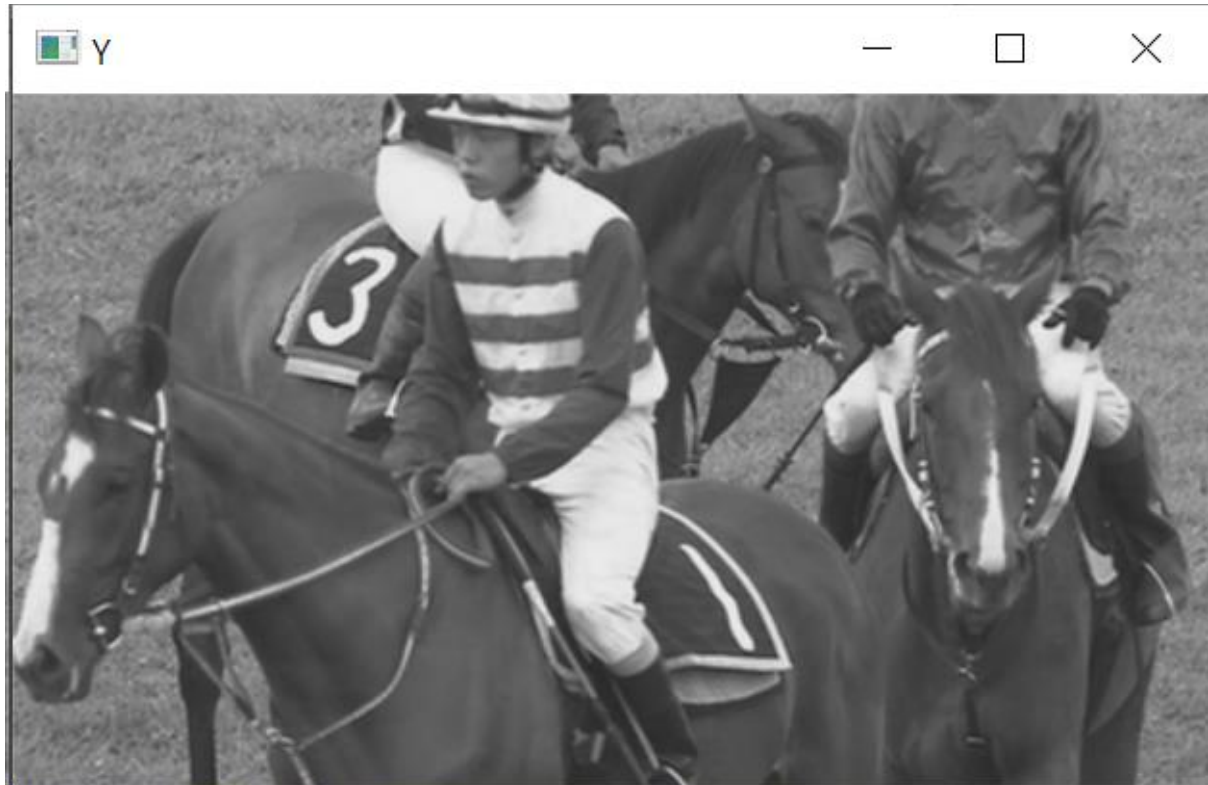


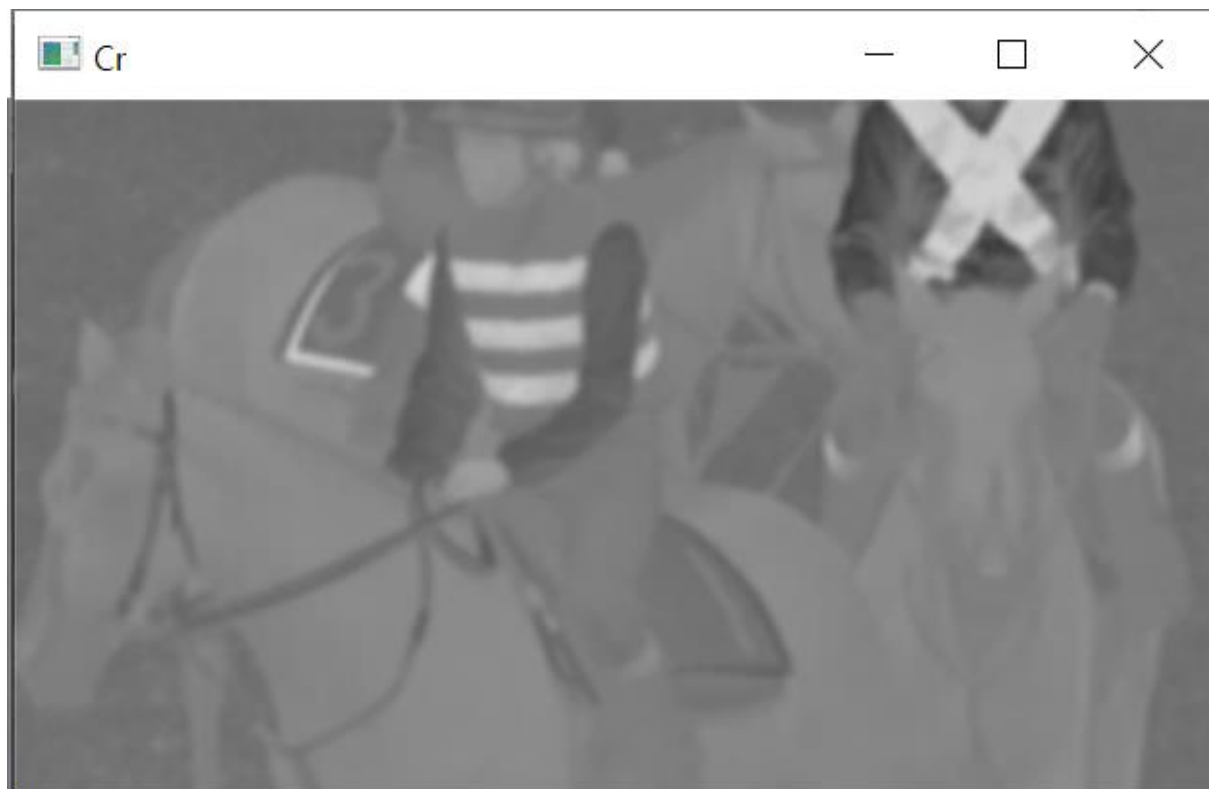
해당 결과를 PYUV로 열면 다음과 같은 결과가 나온다.



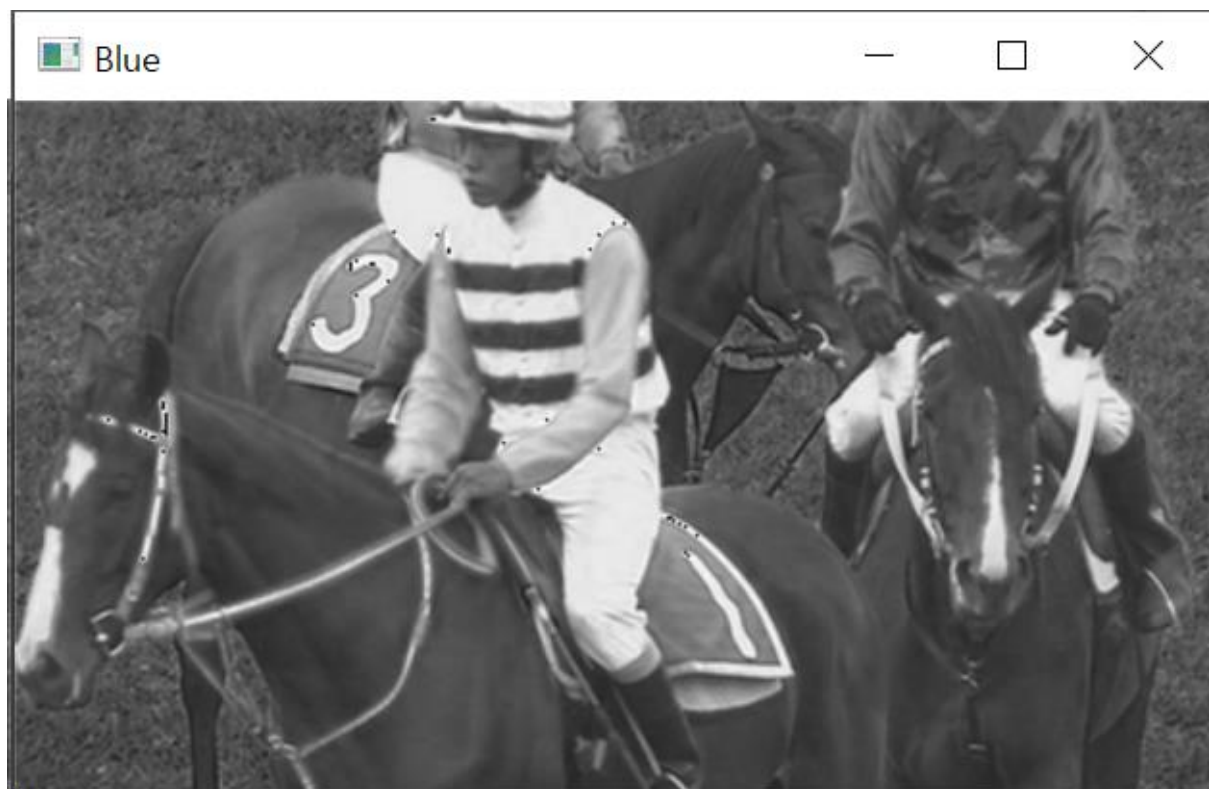
2번 ASSIGNMENT

다음은 y, CB, Cr 데이터를 각각 추출한 결과이다. Cr의 데이터의 경우 빨간색 부분이 상대적으로 밝게 나오는 것을 확인할 수 있다.





다음은 r , g , b 로 변환했을 때의 결과이다. 마찬가지로 red 의 경우 빨간색 부분이 상대적으로 밝게 나오는 것을 확인할 수 있다.



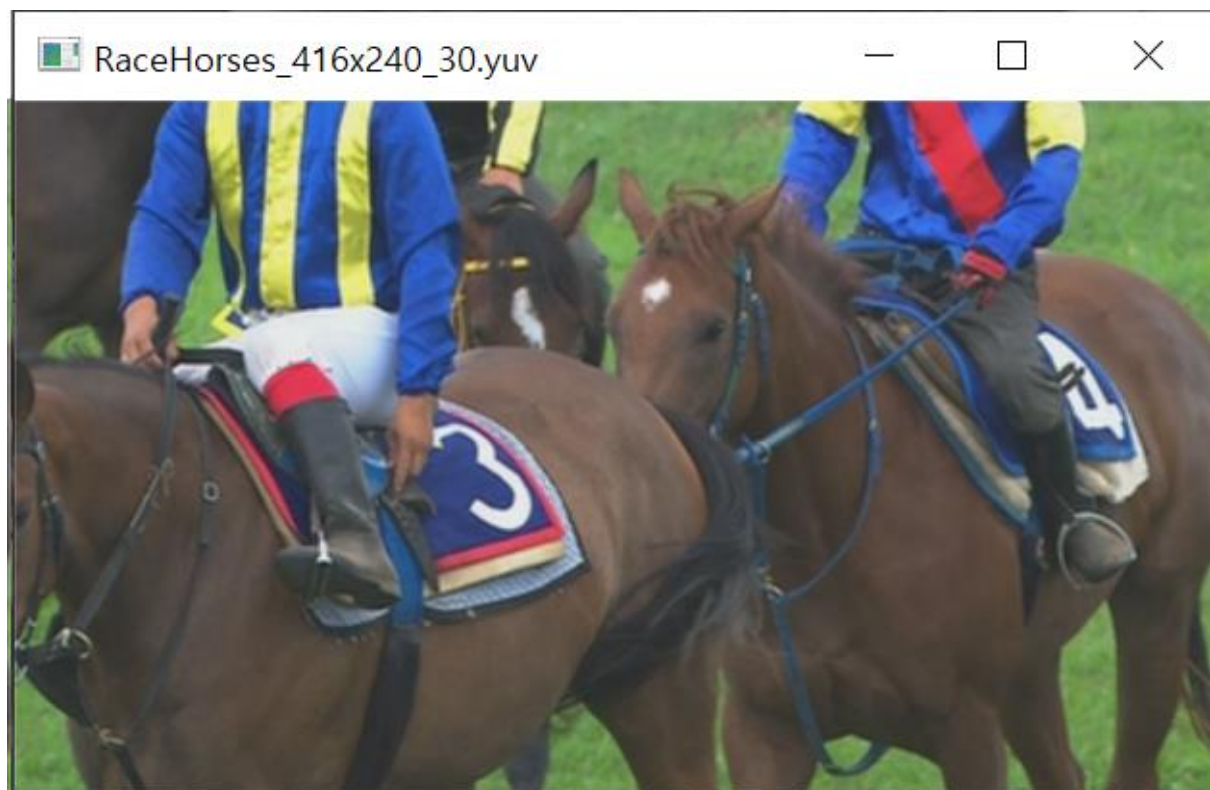
 Green



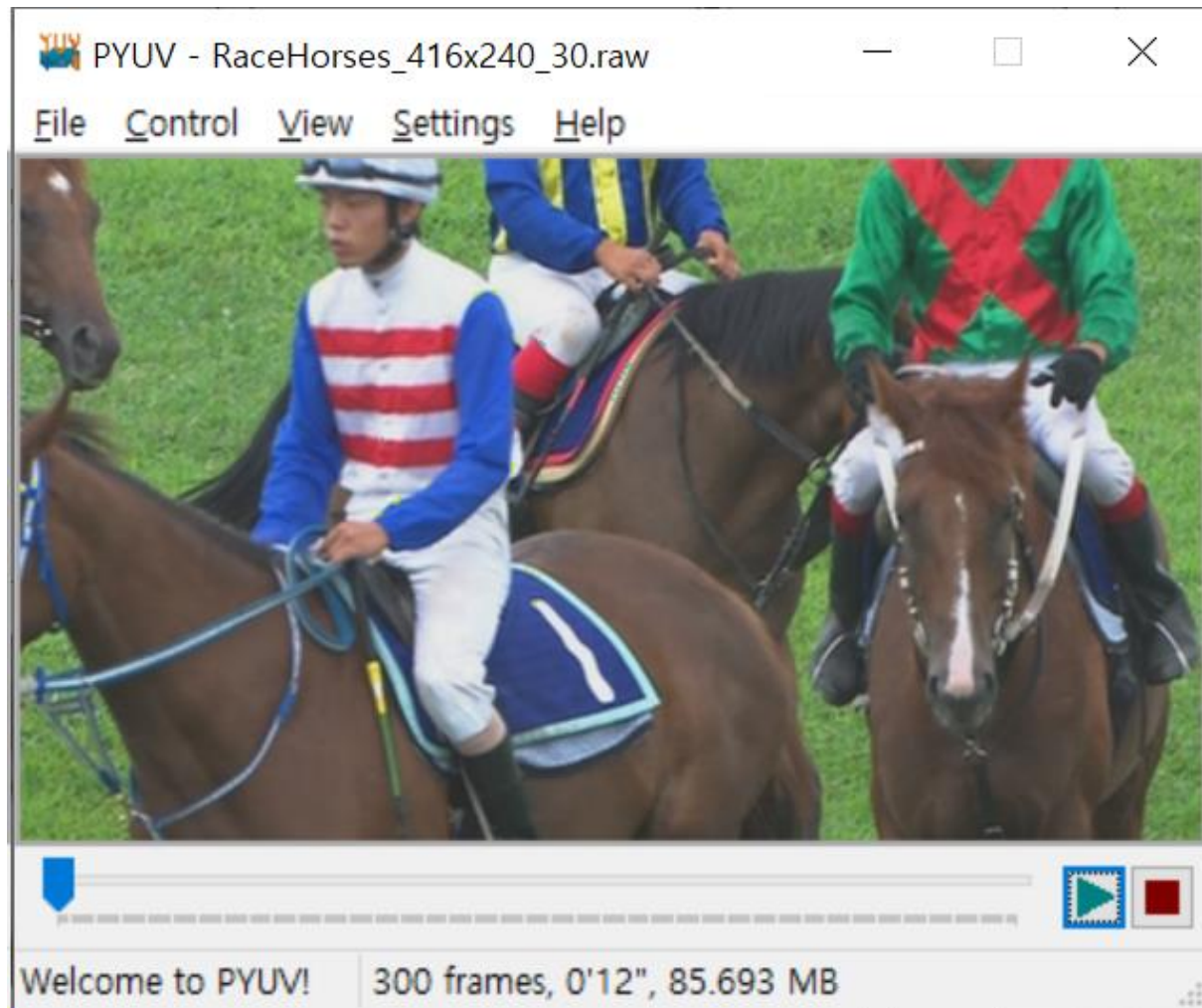
 Red



최종적으로 출력했을 때의 결과이다.



저장된 결과를 pyuv로 출력했을 때의 결과이다.



4. 고찰

해당 과제를 진행하면서 어려웠던 점은 크게 3가지였다.

1번 과제를 하면서 rgb파일을 읽어오는 것에 어려움을 겪었다. Raw 파일은 imshow로 읽어올 수 없기 때문에 fread를 사용하여 읽어야 했고, 이때 r, g, b 데이터가 따로따로 존재한다는 것을 알지 못해 어려움을 겪었다. 하지만 fread 함수를 이용하여 데이터를 읽어오면 r을 먼저 읽어오고, 포인터가 이동하여 g 데이터를 읽어오고, 다시 또 포인터가 이동하여 b 데이터를 읽어온다는 것을 통해 정상적으로 파일을 읽어올 수 있었다.

2번 과제를 하면서 merge를 사용해 영상을 출력할 때 색이 깨지거나 반전되는 문제가 발생하였다. 이 부분은 r, g, b를 각각 담은 배열을 merge할 때 0번지부터 b, g, r데이터를 차례대로 넣고 merge하여야 하는데, r, g, b 순서로 넣어서 발생한 문제였다. 이를 통해 r, g, b데이터를 읽어올 때는 r, g, b순서대로 읽어오지만, 저장할 때는 b, g, r 순서로 저장해야한다는 것을 알게 되었다.

마지막으로 raw, yuv 파일로 저장할 때 4:2:0 압축 파일의 경우 각각의 데이터의 크기가 달라서

어떻게 저장해야하는지에 대해 어려움을 겪었는데, 앞서 rgb를 읽어올 때를 반대로 생각하여 b를 먼저 fwrite를 통해 쓰고, g, r데이터를 순서대로 쓰는 방식을 통해 정상적으로 저장할 수 있었다.

이번 과제를 통해 r, g, b 데이터의 특성에 대해 알 수 있었고, 이를 통해 영상의 취득 방법에 대해 잘 알게 되었다.