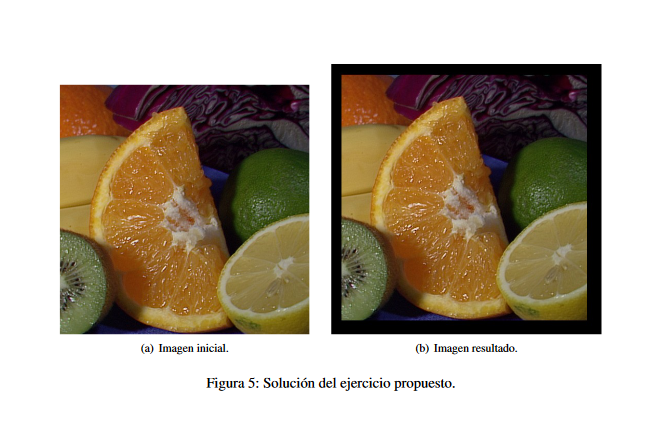
**2. Mostrar una nueva imagen más grande que fruits, en la que se encuentre la imagen fruits entrada a lo ancho y a lo alto. puntos donde no haya imagen serán negros. El resultado debe ser el que indica la figura 5**.



#include <stdio.h>

#include <stdlib.h>

#include <opencv/cv.h>

#include <opencv/highgui.h>

int main(int argc, char\*\* argv) {

if (argc != 2) {

printf("Usage: %s image\_file\_name\n", argv[0]);

return EXIT\_FAILURE;

}

int x = 100;

int y = 200;

//CV\_LOAD\_IMAGE\_COLOR = 1 forces the resultant IplImage to be colour.

//CV\_LOAD\_IMAGE\_GRAYSCALE = 0 forces a greyscale IplImage.

//CV\_LOAD\_IMAGE\_UNCHANGED = -1

IplImage\* Img1 = cvLoadImage(argv[1], CV\_LOAD\_IMAGE\_UNCHANGED);

IplImage\* ImgBig = cvCreateImage(cvSize(Img1->width+x,Img1->height+y),Img1->depth,Img1->nChannels);

// Always check if the program can find a file

if (!Img1) {

printf("Error: fichero %s no leido\n", argv[1]);

return EXIT\_FAILURE;

}

// a visualization window is created with title 'image'

cvNamedWindow("big", CV\_WINDOW\_NORMAL);

// img is shown in 'image' window

cvShowImage("big", ImgBig);

cvWaitKey(0);

for(int fila = 0; fila<Img1->height;fila++){

char \*pImg1 = Img1->imageData+(fila\*Img1->widthStep);

char \*pImgBig = ImgBig->imageData+((ImgBig->height-Img1->height)/2\*ImgBig->widthStep)+((ImgBig->width-Img1->width)/2\*ImgBig->nChannels)+(fila\*ImgBig->widthStep);

for(int col = 0; col<Img1->width;col++){

for(int canal = 0; canal<ImgBig->nChannels;canal++){

\*pImgBig=\*pImg1;

pImgBig++;

pImg1++;

}

}

cvShowImage("big",ImgBig);

cvWaitKey(10);

}

cvWaitKey(0);

// memory release for img before exiting the application

cvReleaseImage(&Img1);

// Self-explanatory

cvDestroyWindow(argv[1]);

return EXIT\_SUCCESS;

}