

AppleIsolator\_for\_jpeg.txt

```
// This ImageJ macro reassigns the background colors to (0,0,0) for all jpeg images
// within a directory. The background must be a low-saturation color, such
// as gray, the foreground images must be relatively high in saturation, and light
// conditions must be uniform. A dialog box appears to ask
// the location of the directory of apple images. The macro requires the plugins
// "Color Space Converter" and "Image Calculator Plus," both of which are available
// at http://rsbweb.nih.gov/ij/plugins/index.html. To change the threshold values
// for apple isolation from the background, simply change the values in the
// changeValues() function below. The default threshold is a saturation level of
// 95. All values below 95 are assigned a value of (0,0,0), while all values above
// 95 retain their original values. The macro was developed to isolate apple images
// (high-saturation) from a neutral (low-saturation) background but could be used
// for any image isolation with similar requirements.
```

```
macro "Apple Image Isolator" {
    dir = getDirectory("Choose a source directory that contains
//apple images");
    list = getFileList(dir);
    setBatchMode(true);
    for (i=0; i<list.length; i++) {
        path = dir+list[i];
        open(path);
        run("Duplicate...", path);//create duplicate file for
//masking
//run("Stack to RGB");// necessary if you are starting
//with a .tiff image
        run("Color Space Converter", "from=RGB to=HSB white=D65");
//process the mask
        setRGBWeights(0, 1, 0);
        setRGBWeights(0, 1, 0);
        run("8-bit");
        changeValues(0, 95, 0);
        changeValues(95, 255, 1);//converts image file to
//simple image containing values 1 for apple area
//and 0 for background
        run("RGB Color");//convert mask to RGB
        img1=getTitle();
        open(path);//opens original image
        img2= getTitle();//next line multiplies the mask
//by the original image
        run("Calculator Plus", "i1=img1 i2=img2 operation=[Multiply:
i2 = (i1*i2) x k1 + k2] k1=1 k2=0 create");
        dotIndex = lastIndexOf(path, ".");
        if (dotIndex!=-1)
            path = substring(path, 0, dotIndex); // remove extension
//to create uncluttered file name
            saveAs("Tiff", path+"clipped");
        close();
        close();
        close();
        close();
    }
}
```