# Report: Multispectral Imaging Binarization



## 1. Introduction

Multispectral Imaging (MS) is one of the most widely used methods for historical document analysis. In the past, many different engineered features have been proposed for the MSI problem. However, this analysis technique ensures that the documents are not damaged. In fact, the investigation is carried out using together ultraviolet, infrared and visible light.

Images acquired with multispectral sensor contain much higher spectral resolution, this is advantages for image analysis, because each image contain different information. This technique enables conservators to obtain valuable information from an ancient document without ruining the material and the paper itself.

The purpose of the work is to do binarization of MSI, with my work I decided to exploit a particular neural network (U-Net). In fact, considering that in recent years the rise of deep learning methods in document analysis has led to more and more deep learning based image binarization approaches.

The use of a network like RNN for example consent to remember previously seen inputs, which enable to make better binarization decision by recognizing the relationship between pixels further apart from each other.

After testing the network on different inputs, I compared this approach with those used in the predecessor exercise.

## 2. Dataset Description

The data are images captured with different wavelength, the wavelengths are captured with specific instrument capable of frequencies beyond the visible light range.

The test dataset consist of some multispectral (MS) images generated from historical text. For every image we have different wavelength,  taken from a range of 240, 1100 wavelengths.

| Name | Wavelength | Light |
|---|---|---|
| F1s.png | 340 | UV |
| F2s.png | 500 | Visible 1 |

| Name | Wavelength | Light |
|---|---|---|
| F3s.png | 600 | Visible 2 |
| F4s.png | 700 | Visible 3 |
| F5s.png | 800 | IR 1 |
| F6s.png | 900 | IR 2 |
| F7s.png | 1000 | IR 3 |
| F8s.png | 1100 | IR 4 |

This kind of light beyond the visible spectrum reveal features that  cannot be seen by the human eye. The wavelengths considered are the Infrared wavelengths and the Ultraviolet wavelengths, both of this approach gives different information about the images.

**Visible Wavelengths**

The visible spectrum is the portion of the electromagnetic spectrum that is visible to the human eye. Usually, the human eye can recognize wavelengths from about 380 to about 750 nanometers.

**InfraRed Wavelengths**

Infrared radiation, is a region of the electromagnetic radiation spectrum where wavelengths range from about 700 nanometers to 1 millimeters. Infrared is used in a variety of applications, it is usually used to see through darkening or charring.

**Ultraviolet wavelengths**

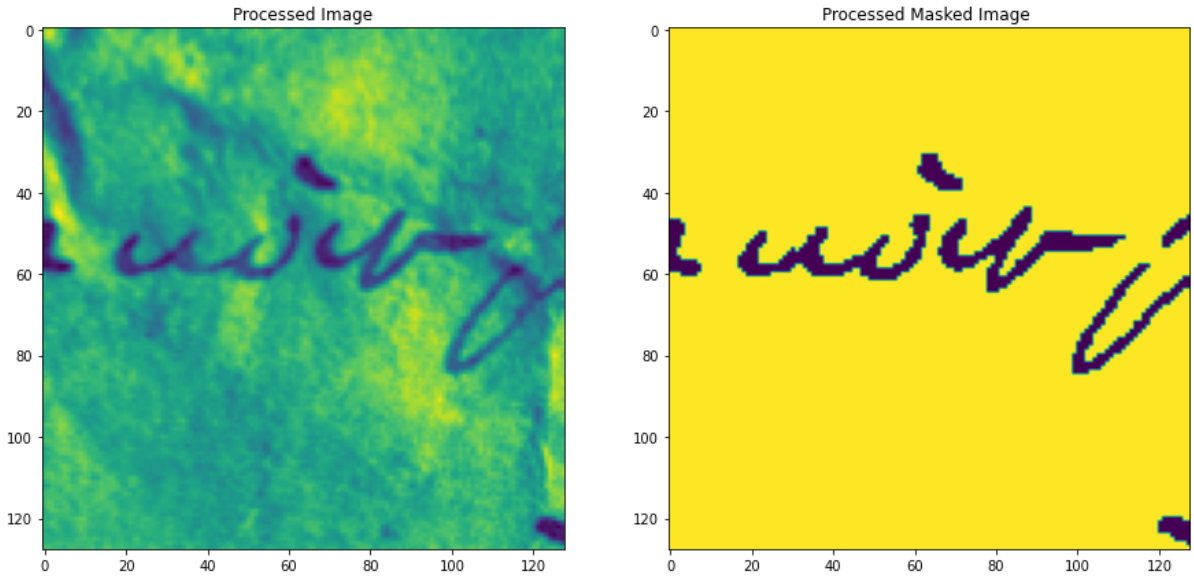
Ultraviolet radiation is the portion of the electromagnetic spectrum end of the visible light range to the X-ray region. Ultraviolet radiation lies between wavelengths of about 400 nanometers. This wavelength enhances visibility of faded.

# 3. Background

## 3.1 Architecture

In my approach, I divided the images to be binarized into non-overlapping block of $128 \times 128$ pixels, padding the images with zero if necessary. These blocks are then converted into input sequences to the U-Net and binarized separately from each other.

I chose this approach to facilitate the work of the network, which needs input with the same size all the time. Also, through this mechanism, I greatly increased the number of images to be used to train the network.




The peculiarity of this choice is that we will almost unintentionally force the model to classify a pixel by considering only a limited number of pixels close to the one under consideration, that is, those that fall within the same block.

The choice to make the pad is because I wanted each image to be divisible in the 128x128 block.

## 3.2 U-Net

**U-Net** is a convolutional neural network used to do image segmentation. It consists of two paths:

- Contracting path: This path follow the typical architecture of a CNN, it is the repeated application of two 3x3 convolutions, each followed by a RELU and a 2x2 max pooling operation. During this phase, the spatial information is reduced while features information is increased.
- Expensive path: this path combines the feature and spatial information through a sequence of up-convolutions and concatenations with high resolution features from the contracting path. At the end there is a final layer, a 1x1 convolution is used to map each component feature vector to the desired number of classes.

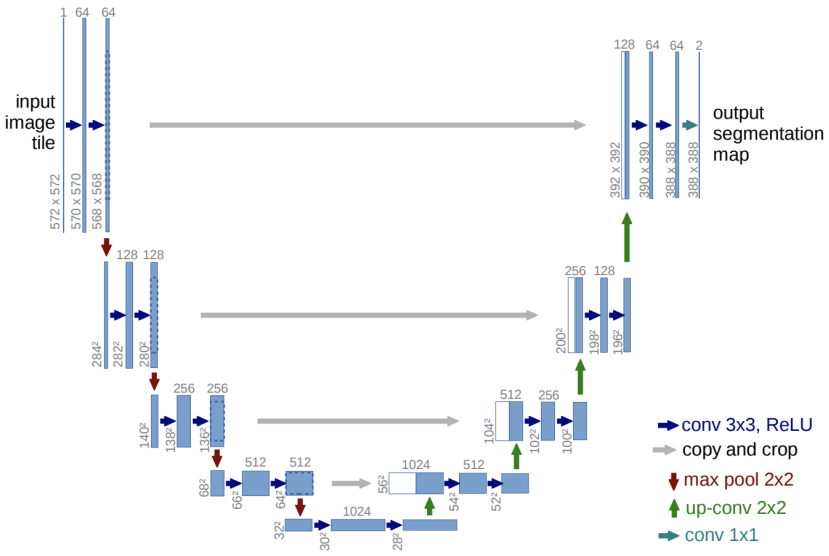



**Fig. 1.** U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

*(image taken from: Link)*

# 4. Methods

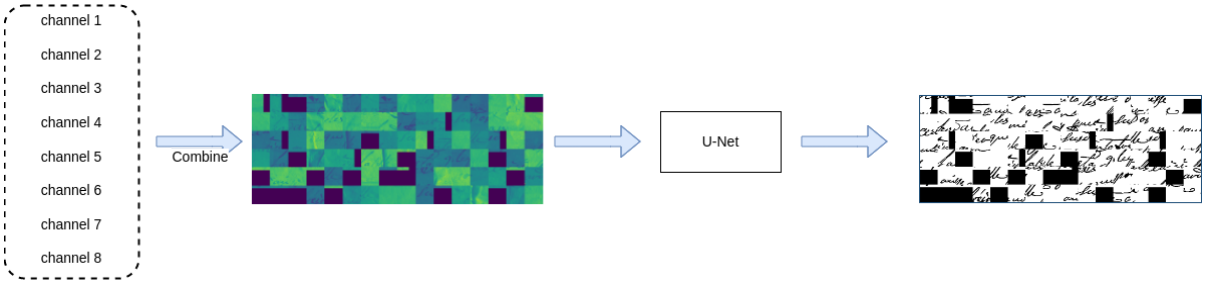
As mentioned earlier, I used U-Net as my basic neural network. After that, I decided to do several experiments to combine the information contained by the different wavelengths.

I then followed two approaches, in the first I combined the contained information from the different wavelengths, and trained the model to evaluate its behavior. After that, I trained the model on images with only one channel to evaluate whether the contribution of the different channels was capable of increasing the performance of the model.

It should be noted that the model training was done on a google colab notebook, which I add to the code directory for completeness.

## 4.1 Approach 1 - Combination channel

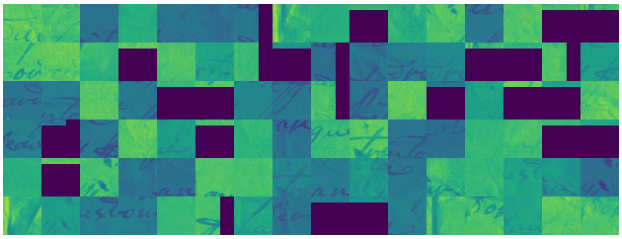
### 4.1.1 Dataset



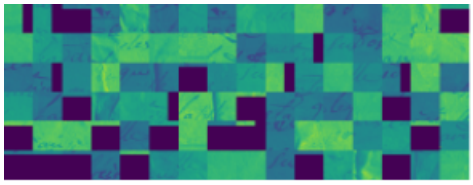

In this approach, the input to the model was created by combining all channels for each individual image. Thus, given an image with 8 channels, 8 different images were randomly created by combining 64x64

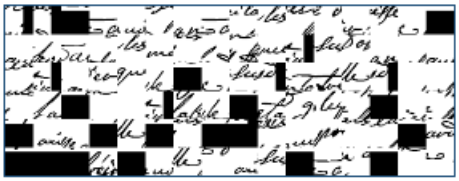
size blocks of the images corresponding to the various channels.

At the end of the process, the created images similar to those shown in the figure below:



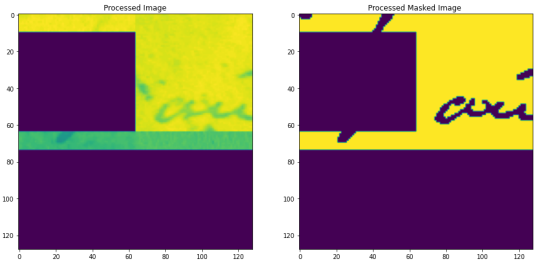
Following the same order in which I created a new image by combining the various channels, I also created new GT images. In this way, I passed a pair of mirror images to the model.





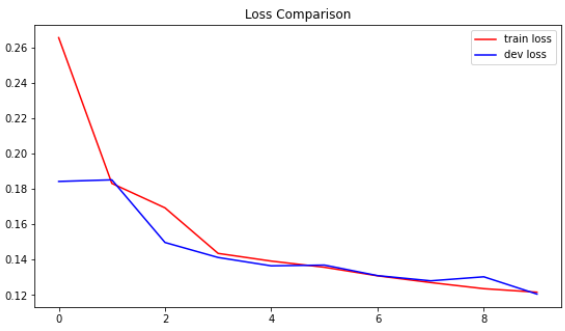
Before dividing the images into blocks and then reassembling them, I had to pad them so that the images were a multiple of the size of the blocks. That is why there are some black blocks in the images.

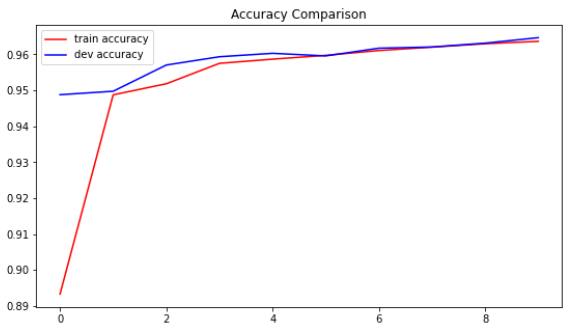
After this step before passing the created image to the model I still divided the image into blocks of size 128x128.




## 4.1.2 Results

In a first test, I trained the network for 10 epochs. As can be seen from the graph below, under these circumstances the model was already performing quite well with an accuracy well over 90% and a loss of less than 15%

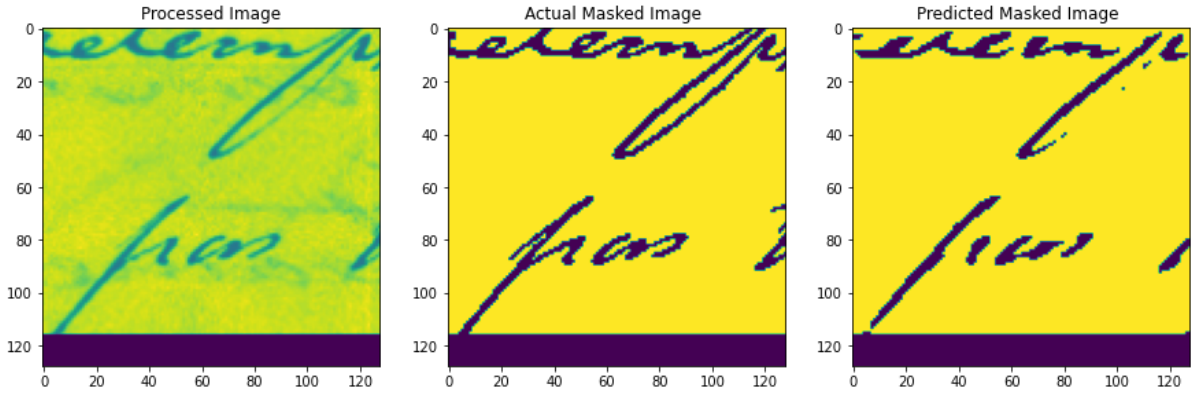


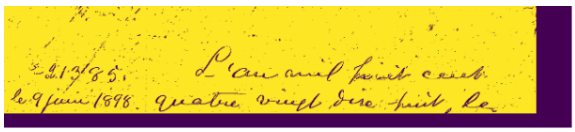
Also on the validation set I obtained good results:

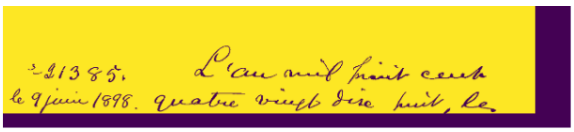
> loss: 0.1205 - accuracy: 0.9646

Taking a look at the predicted images and comparing them with the input images and the gt, we can see right away that although the model can all in all recognize the image broadly, the image created is still noisy with some pixels misclassified.




Looking in full at the behavior of the model on new images, it becomes more apparent how the model do not classify each pixel correctly.





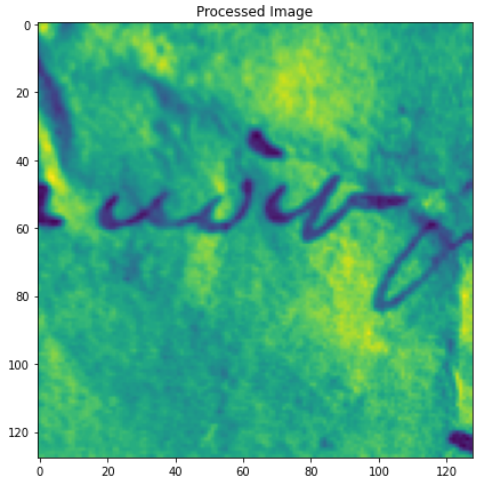
So despite some imperfections, I noticed how nevertheless under these circumstances the network performed quite well in classification. This will become clearer later when I mention the results of the metrics.

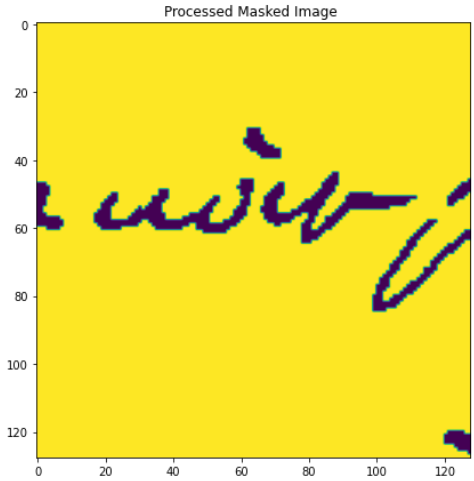
## 4.2 Approach 2 - One channel approach

### 4.2.1 Dataset

In this second approach, instead of combining the information provided by the different channels, I used a single visible channel.

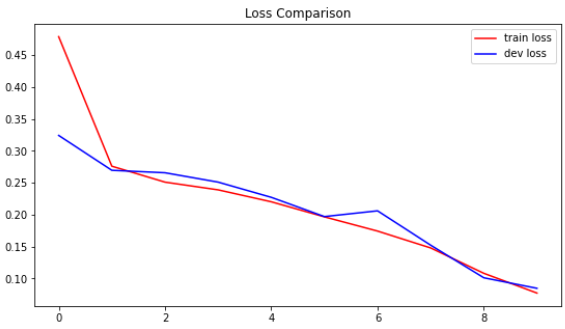
Even in this case, in order to run the model I separated the images in blocks of size 128x128.

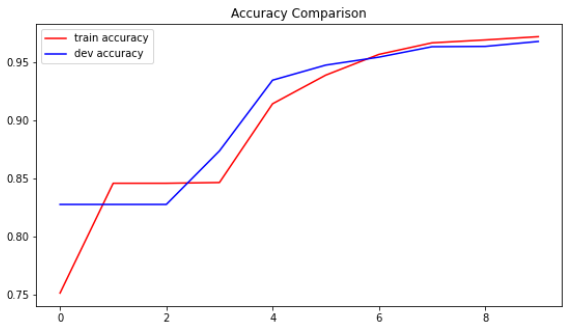



### 4.2.2 Results

Also for this approach I trained the model for 10 epochs. The surprising thing is that in this case with only 10 epochs the model performed almost as well as the previous version with the same number of epochs.



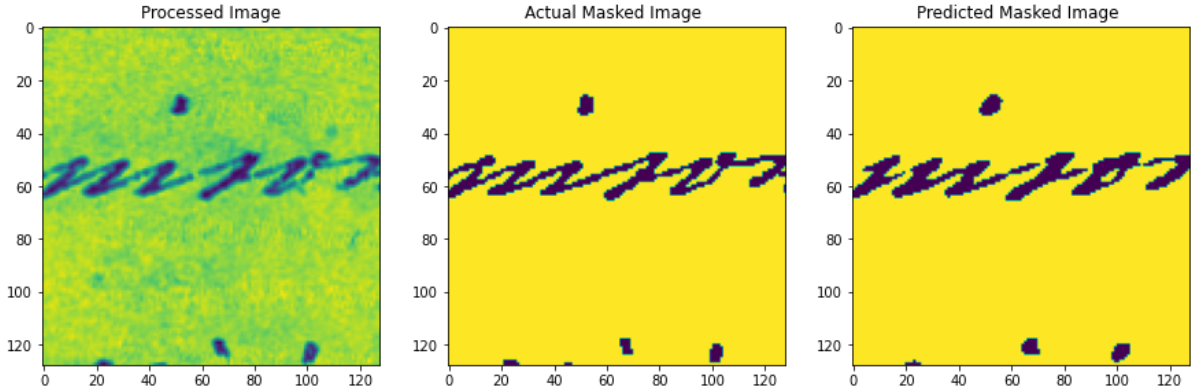




The accuracy values are more or less similar, however in this case an even lower loss value is achieved.

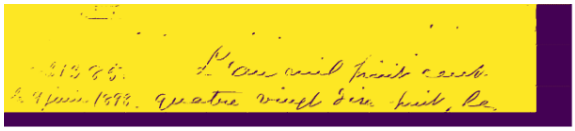
> On the validation set the results are:

loss: 0.0847 - accuracy: 0.9682

The generated images are less noisy and clearer.




If we take a look at the full image:



Though again the image is not completely sharp and clear, a better classification can still be recognized.

# 5. Comparison and Conclusion

To evaluate the performance of the model, I compared the results obtained in the two approaches tested with those obtained in the previous exercise (i.e., Su.et al and Otsu).

For details on the previous approaches and the measurement techniques used, I refer to the description of the previous exercise.

It should be pointed out that before calculating the DIBCO measurements on the results obtained from the two tested approaches, I removed the padding values that I had entered in the previous to change the image size. In this way, I am more confident of the veracity of the information obtained.

The results of the comparison can be seen on the following table:

| Method | approach 1 (10 epochs) | approach 2 (10 epochs) | Otsu | Su et al |
|---|---|---|---|---|
| F-measure | 0.85 | 0.826 | 0.71 | 0.57 |
| PSNR | 45.53 | 41.649 | 32.89 | 26.41 |
| DRD | 0.0125 | 0.0083 | 8.24e-05 | 5.77e-05 |

It can already be seen that the accuracy of the new approaches is far better than the previously implemented approaches. Also confirming this view are the results obtained with the other measures. Thus, the approaches tested in this exercise present much more accurate results with a lower level of bias.

We can also see that the approach obtained by combining different channels is still better than the other. In fact, the values of the metrics are higher in the first approach. This is probably due to the fact that the first approach having more images at its disposal succeeds better than the other in recognizing the characteristics of the image in question, and thus in classifying the pixels correctly.

However, some factors must be taken into consideration when analyzing the results obtained, although in fact the results obtained with the first approach are better this is also due to the fact that the first model was trained on a larger number of images. In addition, one must also take into consideration that among the channels considered to create the images for the first approach I did not remove the "IR4" channel. In my opinion this choice penalized the results obtained because the model considered this channel as a normal image, and because of the very content of the image I do not think that learned important features.

In addition, these models are usually trained for a greater number of epochs, so I believe that simply doubling the number of epochs yields significantly better results for both approaches.

Finally, it seems quite clear all things considered that in circumstances where information from different channels is combined, better results are obtained.