Normalized Artificial Intelligence Labeling System

= N-A.I.-LS =

to identify hidden signals from fingernails

User's Guide

version 1.0 - August 2024

by E. Canessa & L. Tenze

This is a Python-based algorithm that allows to identify hidden, temporal signals in the variation of colors from fingernails. It is based on the nails segmentation using deep learning models. Healthy fingernails have a uniform color and are visually smooth. However, as one ages, nails may become more brittle and may also present discoloration due to injury, fungal and viral infections, medications, etc. Furthermore, the appearance of the fingernails can also change due to some medical conditions.

https://github.com/canessae/nails

SECTION 1

Fingernails colors via *N-A.I.-LS*

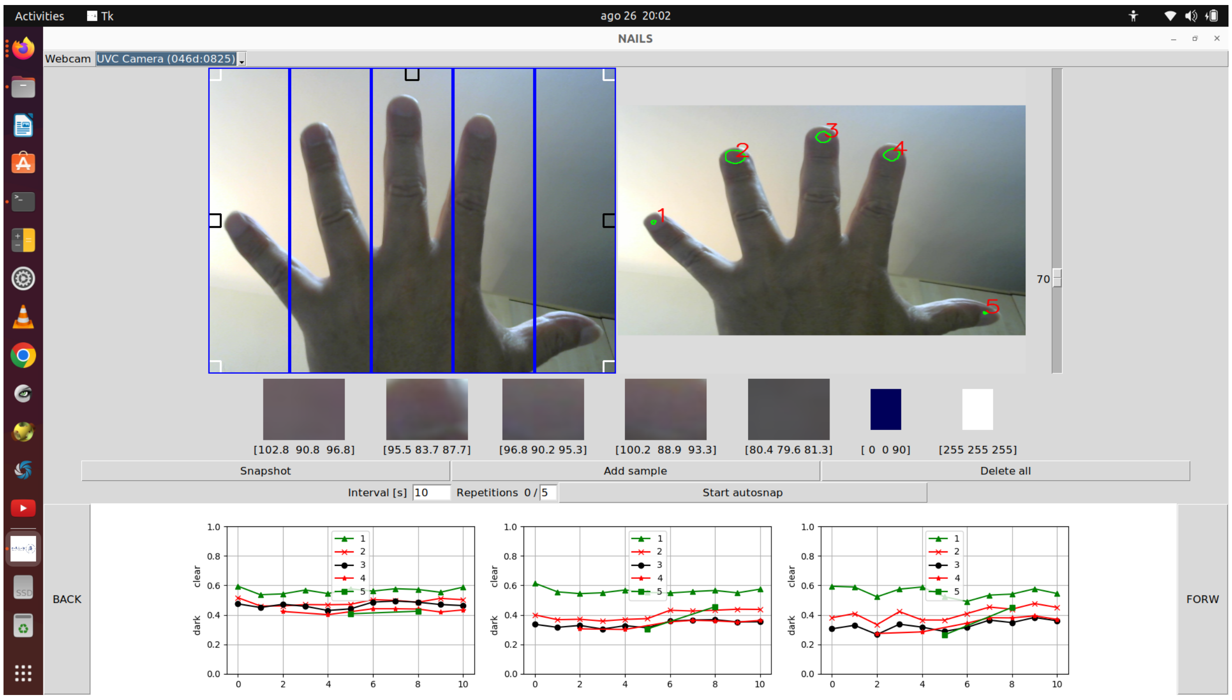


Fig.1: *GUI for the Normalized Artificial Intelligence Labeling System (N-A.I.-LS) aiming to identify hidden signals from fingernails.*

The Graphics User Interface (GUI) interface of the *N-A.I.-LS* package is based on the tkinter python library. Some other libraries have been integrated in order to add plot history (with the matplotlib and numpy libraries being used), to show the real-time and snapshots acquired from the camera (`opencv-python`). Some classes have been developed in order to solve other tasks such as videocapture pause and restart, complex processing of the acquired images and so on.

In particular the CameraSearch library has been developed in order to enumerate and get info about the connected webcams: no one library completely functional has been found from the open source solutions. This library is based on the `v4l-utils` package; hence the current solution is not portable to Windows or MacOS systems.

In order to make the library working, the user needs to install the `v4l-utils` package, which can enumerate and detect many aspects of installed cameras. This class has been mainly used to implement the camera combo box to allow the user change the current acquisition camera.

When the user gets a snapshot from the video source showing his/her open hand (subdivided in five equally rectangular areas separated by blue lines as in the input video on the left of Fig.1), the system:

- Search for, and processes, the Black and White regions in order to normalize the image pixels (I_x). The algorithm then normalizes the R(ed), G(green) and B(lue) values at pixel x according to this equation:

$$\eta_x = \frac{I_x - Black_x}{White_x - Black_x} \quad , \tag{1}$$

where x can be R, G or B region displayed on the left video.

- The referential Black value is evaluated taking into account the minimum value inside the small squared regions located at the four corners of the input video. The referential White value is evaluated taking into account the maximum value inside the small squared regions located at the three central areas of the input video.
- Then the Convolution Neural Network is fed by the acquired image in order to get the nails regions. The neural network used has been downloaded from the following link: `https://github.com/ademakdogan/nails_segmentation.git` This network performs better than alternative solutions because it is based on the segmentation concept. The network has been trained over the default data set and the weights are applied to real images with reasonable good results.
- Finally the system gets the RGB values from the nails regions, normalizes the RGB values and stores the RGB normalized values.
- As illustrated in Fig.1, the RGB normalized values of the respective R,G,B curves for each fingernail identified by the algorithm are plotted in the bottom region of the GUI, where the RGB history is displayed.

1.1 Requirements

The minimum hardware needed to test *N-A.I.-LS* is any standard PC Computer *e.g.*, Intel Core i5, 64bit and at least 4G RAM, running a recent release of Linux O.S. (23.04LT or newer) having an internal webcam and Python installed.
In case of using an external USB webcam, the user needs to install the `v4l-utils` package, which can enumerate and detect many aspects of installed cameras. This class has been mainly added to *N-A.I.-LS* in order to implement the camera combo box to allow the user change the current acquisition camera.
Optional hardware that can be used are an external USB webcam with optical zoom, a closed box to insert one hand to make more precise measurements in a luminous controlled environment (resembling the ancient ”Mouth of Truth” attraction in Roma, Italy –shown below).

Fig.2: *3D printed Copy of the ancient Roman marble mask "Mouth of Truth".*

1.2 Copyright, Credits and Contacts

For further information, or to report Bugs, please visit:
https://github.com/canessae/nails

SECTION 2

Install/Usage

The latest version of the *N-A.I.-LS* code can be downloaded from GitHub as

```
git clone https://github.com/canessae/nails.git
```

Then, create a Python virtual environment for the package and install the needed dependencies in the "nails" directory as follows

```
cd nails

python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
deactivate
```

Next, download the "`best_model.pth`" file (183,4 MB) into the "model" directory

```
cd model
```

from the following DropBox single link:

```
https://www.dropbox.com/scl/fo/spk489eavbkggaeplg4sm/
      AMPNp-c0tayeFPsBTvryYMk?rlkey=qm5rnncl7cwdb5fl6pof3c4em
      &e=1&st=gq4a9n64&dl=0
```

The Neural Network (NN) used has been downloaded from `https://github.com/ademakdogan/nails_segmentation.git` We have found that this Convolutional (NN) performs better than alternative solutions because it is based on the nails segmentation using deep learning models, Our NN has been trained over the default data set and the weights are applied to real images with reasonable good results.
To start using the *N-A.I.-LS* application via the "nails.py" application in the "nails" directory, issue the commands

```
cd nails

source venv/bin/activate
python3 nails.py
deactivate
```

N-A.I.-LS GUI

To start using the *N-A.I.-LS* GUI, first select the acquisition webcam from which you will get the video image of the hand (input video on the left of Fig.1). In Fig.3 below, there is a menu listing one internal webcam and a second UVC webcam connected to the PC via USB. The *N-A.I.-LS* package is based on the `v4l-utils` package; hence the current solution is not portable to Windows or MacOS systems.

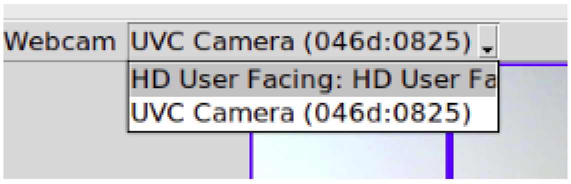


Fig.3: *Selected webcam menu.*

After positioning each finger in one of the five equally rectangular areas separated by blue lines in the input video, there are two modalities to get the nails regions by the Convolution Neural Network.

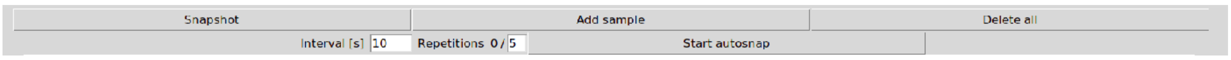


Fig.4: *Modalities to acquire (single or multiple) images for the fingernails regions.*

As illustrated in Fig.4, one modality is to acquire single images for the nails regions by selecting and pressing the "`Snapshot`" button or just pressing the "`Enter key`". After a few seconds you will get the identified areas around the fingernails (output image on the right of Fig.1). Such areas can be regulated in size by the cursor appearing on the right of the GUI as shown for example in Fig.5 below. The second modality, is to acquire a large temporal set of data at given intervals as indicated in Fig.4, by selecting "`Start autosnap`".

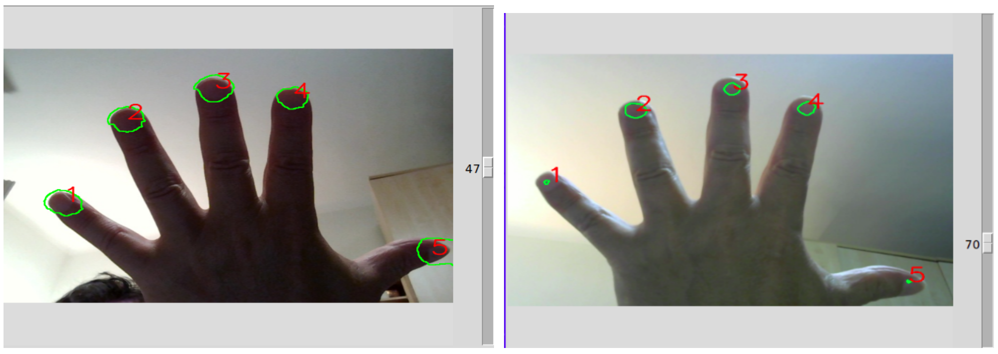


Fig.5: *Fingernail areas regulated in size by the cursor on the right of the GUI.*

Finally the system gets all (single or multiple) RGB values from the fingernails regions, normalizes the RGB values, stores the RGB normalized values from both modalities and plot them in one of the three R-G-B plots shown in Fig.6. As illustrated in this figure, the RGB normalized values of the respective R,G,B curves for each fingernail identified by the algorithm are plotted in the bottom region of the GUI, where the RGB history is displayed.

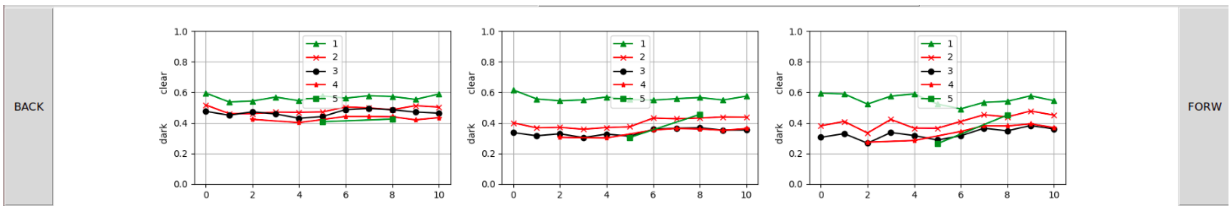


Fig.6: *Red, Green and Blue plots of the mean color values for each finger.*

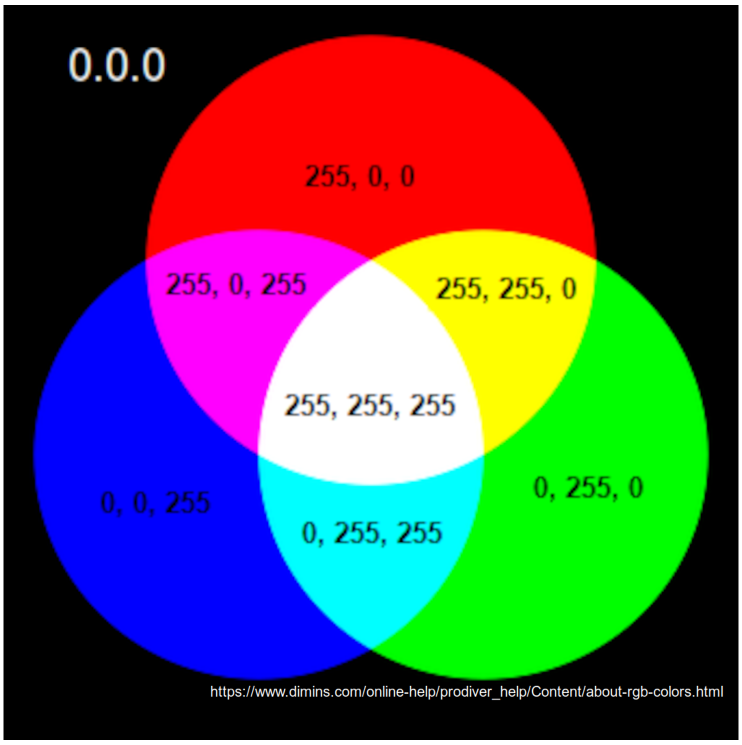
Note: When selecting the ”`Snapshot`” single button, you can ”`add sample`” or press the ”`Space key`” to plot the single RGB values estimated. In all cases the plotted values correspond to a median value obtained for each single fingernail region identified.
By selecting ”`Delete all`” all recorded data is cancel.

SECTION 4

RGB Hidden signals

4.1 RGB Colors

The RGB color model is based on the theory that all visible colors can be created using the primary additive colors of Red, Green and Blue. To each of these colors is assigned a value in the range of 0–255 as in the figure. When these values are combined in different amounts, colors are produced. For example: (0,0,0) is black and (255,255,255) is black.



0.0.0
255, 0, 0
255, 0, 255
255, 255, 0
255, 255, 255
0, 0, 255
0, 255, 0
0, 255, 255
https://www.dimins.com/online-help/prodiver_help/Content/about-rgb-colors.html

Fig.7: *Visible colors recreated using the primary additive colors of Red, Green and Blue.*

4.2 Colors signals

According to literature, nails may reflect our interior health. To some extend, what they may be trying to tell us about our health is briefly depicted in the illustration below.

In all cases, keep in mind that making a definitive diagnosis is up to the Dermatologist specialist and it is not a question of just taking a "look" –laboratory tests are always necessary!

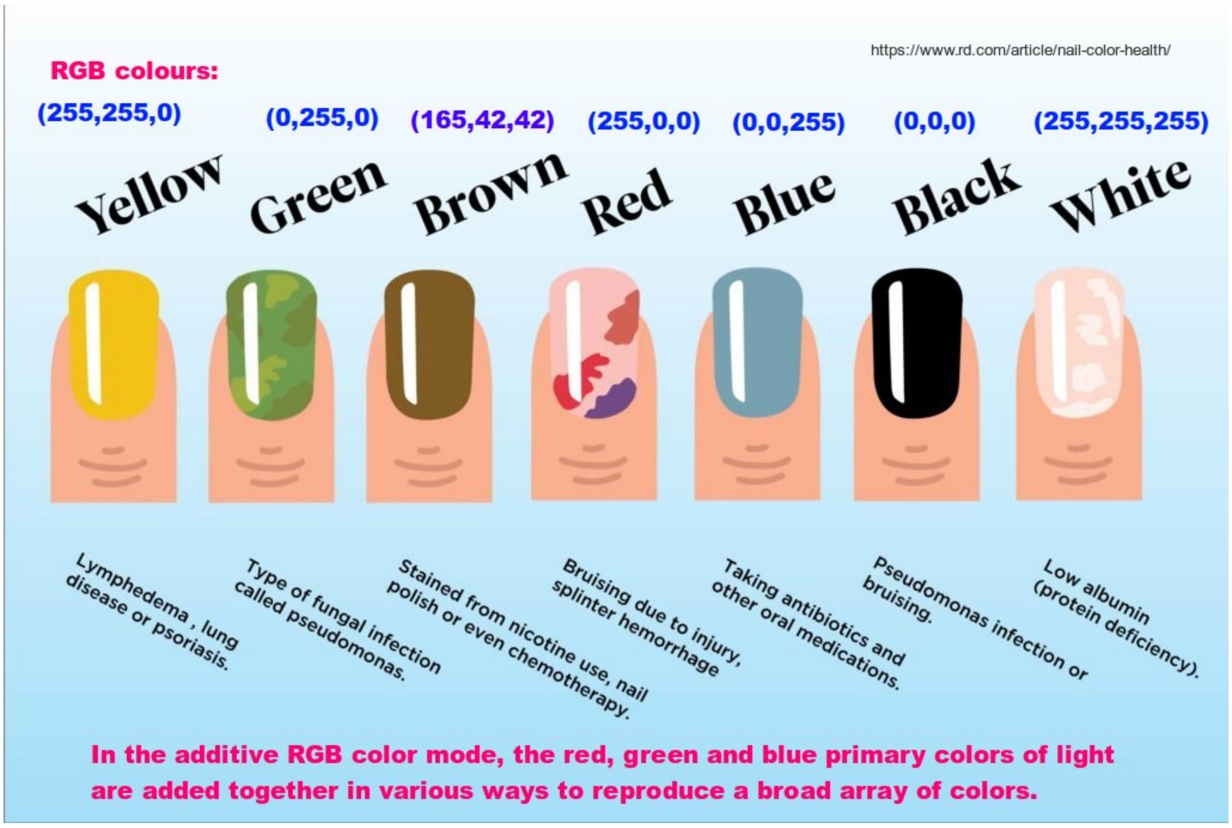

https://www.rd.com/article/nail-color-health/
RGB colours:
(255,255,0)
(0,255,0)
(165,42,42)
(255,0,0)
(0,0,255)
(0,0,0)
(255,255,255)
Yellow
Green
Brown
Red
Blue
Black
White
Lymphedema , lung disease or psoriasis.
Type of fungal infection called pseudomonas.
Stained from nicotine use, nail polish or even chemotherapy.
Bruising due to injury, splinter hemorrhage
Taking antibiotics and other oral medications.
Pseudomonas infection or bruising.
Low albumin (protein deficiency).
In the additive RGB color mode, the red, green and blue primary colors of light are added together in various ways to reproduce a broad array of colors.

Fig.8: *Finger nails colors and health.*