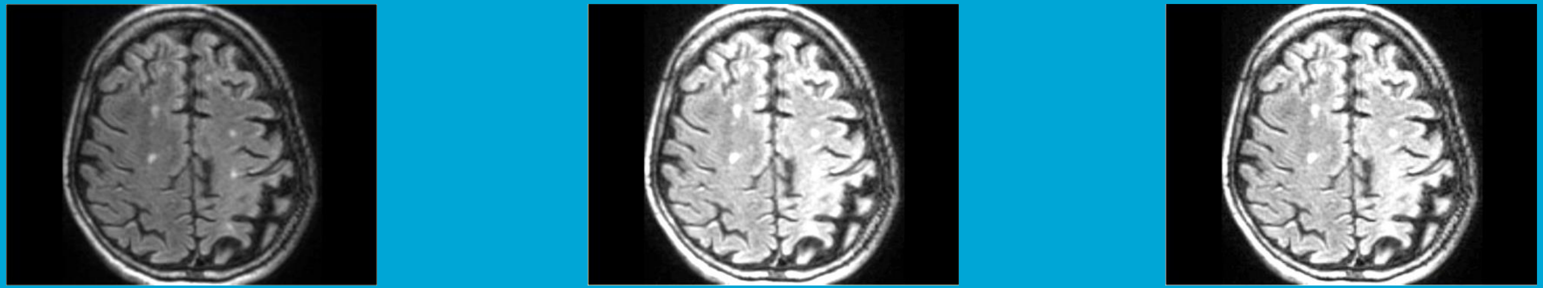
Video Contrast and Sharpness Enhancement

T. Petersen, T. C. Mol, E. Cozza

16/02/2024

Fields of Application

- Satellite Imaging
- Medical Imaging
- AI and Computer Vision applications
- Multimedia applications

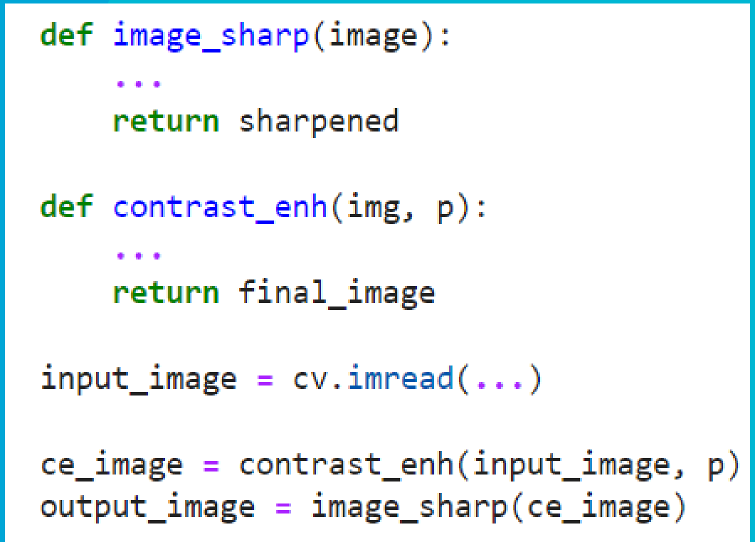


Satellite image MRI

Design Modules

1. *Contrast Enhancement* module
2. Convolution with *sharpener* filter

Software Implementation



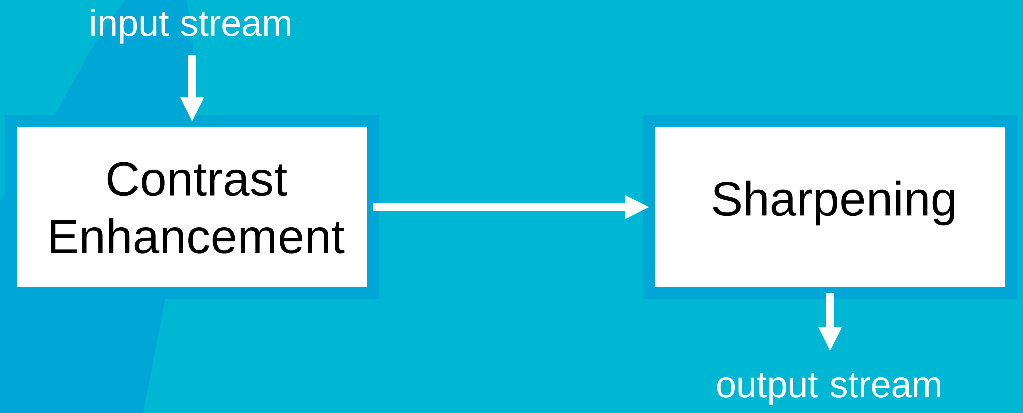
```
def image_sharp(image):
    ...
    return sharpened

def contrast_enh(img, p):
    ...
    return final_image

input_image = cv.imread(...)

ce_image = contrast_enh(input_image, p)
output_image = image_sharp(ce_image)
```

Hardware Implementation



01

Software Implementation

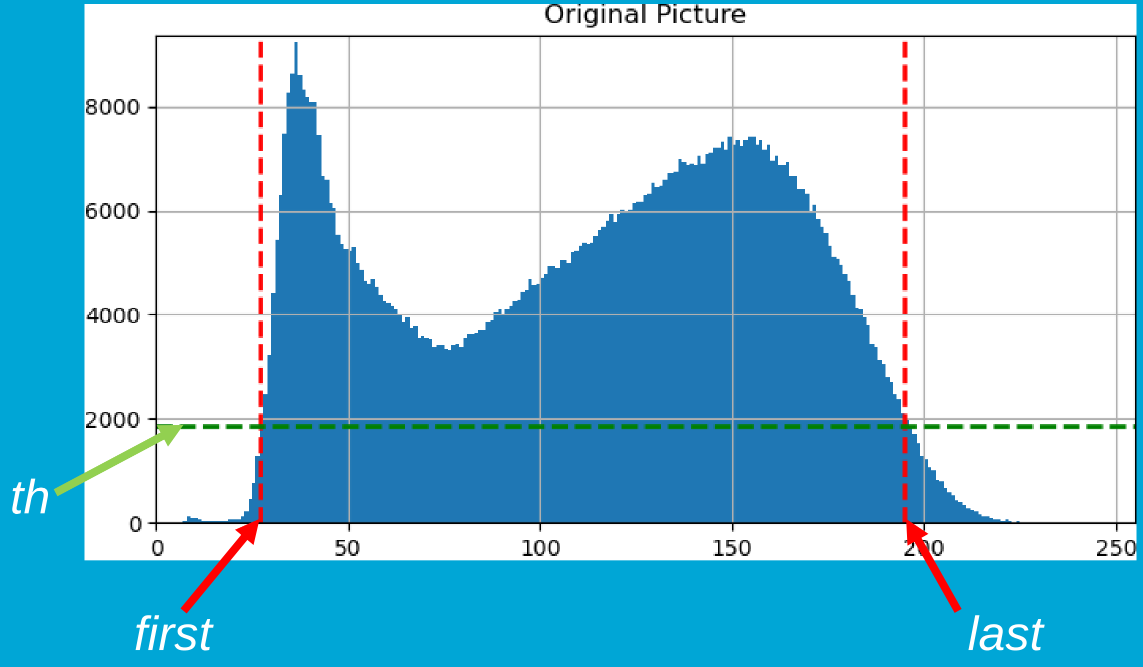
Contrast Enhancement

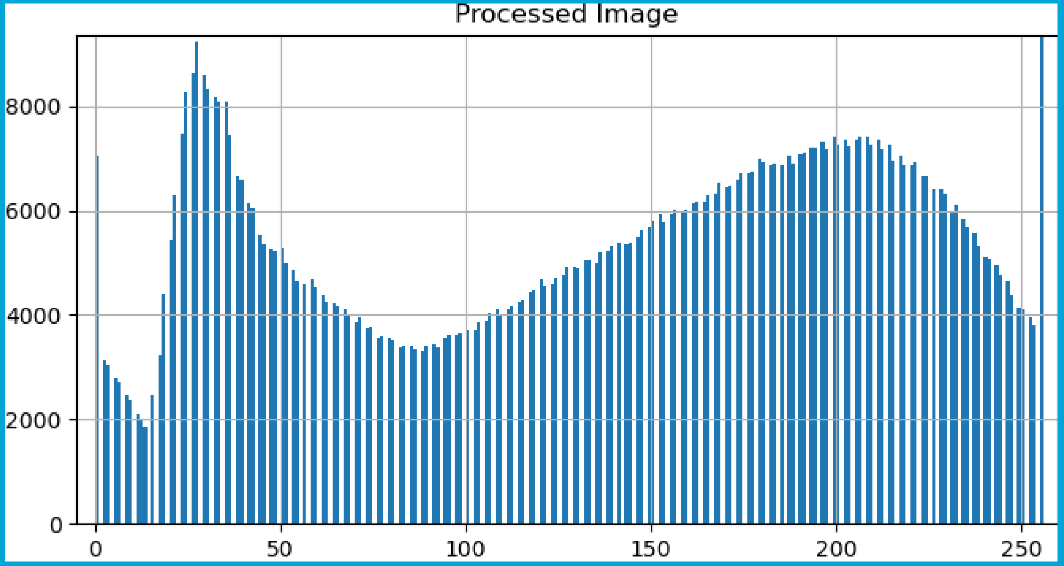
1. Image decomposition into 3 channels
2. Histogram Computation
3. $th = \max(histogram) \cdot p$
4. $k = \dfrac{255}{last - first}$
5. Adjust the brightness level of all the pixels:

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) < \texttt{first} \\ \mathbf{k} \cdot f(x,y) - \texttt{first} & \text{if } f(x,y) \in [\texttt{first}; \texttt{last}] \\ 255 & \text{if } f(x,y) > \texttt{last} \end{cases}$$

6. Repeat steps 2-5 for each channel

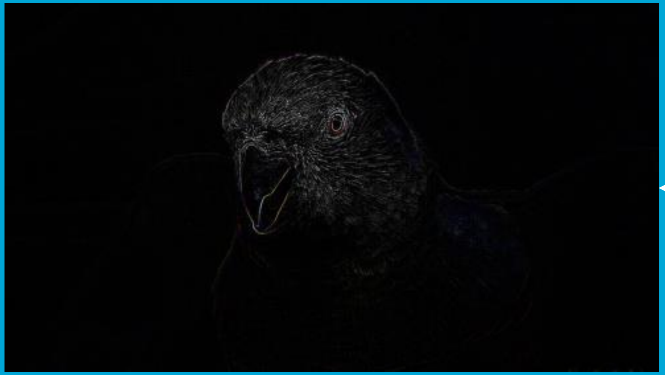
(***p*** is a user-defined input parameter)





$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Identity



$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Edge Detection

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Sharpen

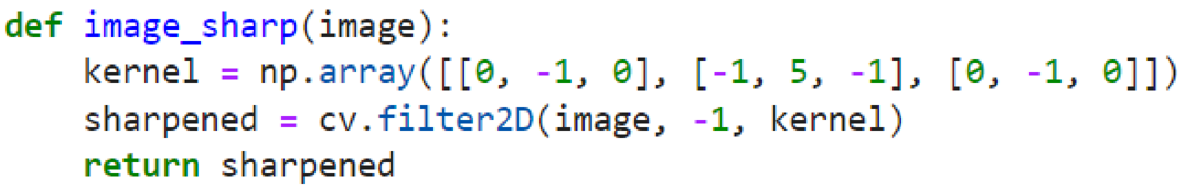
Edge Enhancement

- Based on the *convolution operation* for a fixed kernel

$$g(x,y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \omega(i;j) \cdot f(x+i, y+j)$$

- where:
  - *ω(i;j)* is the sharpener filter
  - *f(x,y)* is the input image pixel in coordinates (x;y)
  - *g(x,y)* is the output image pixel in coordinates (x;y)

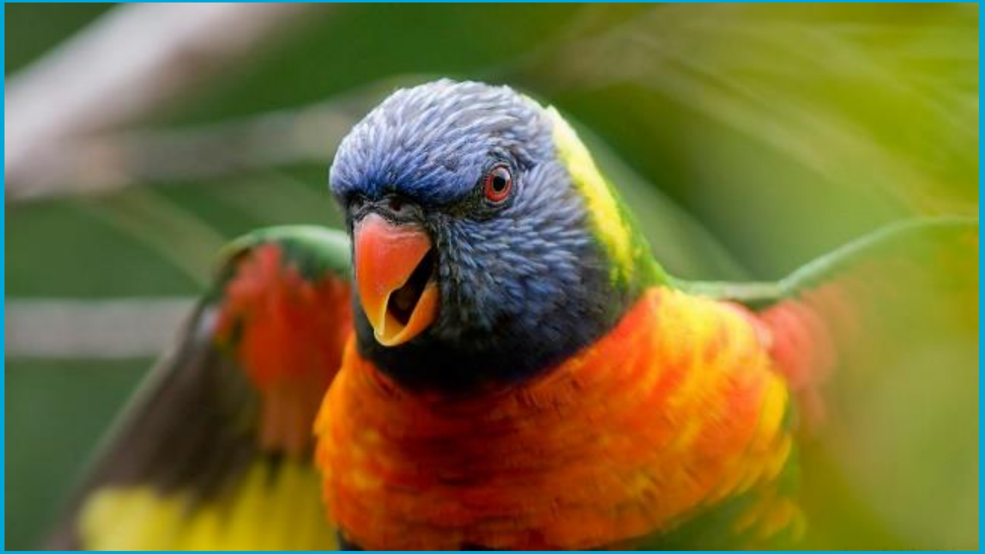
- Sofware Implementation:



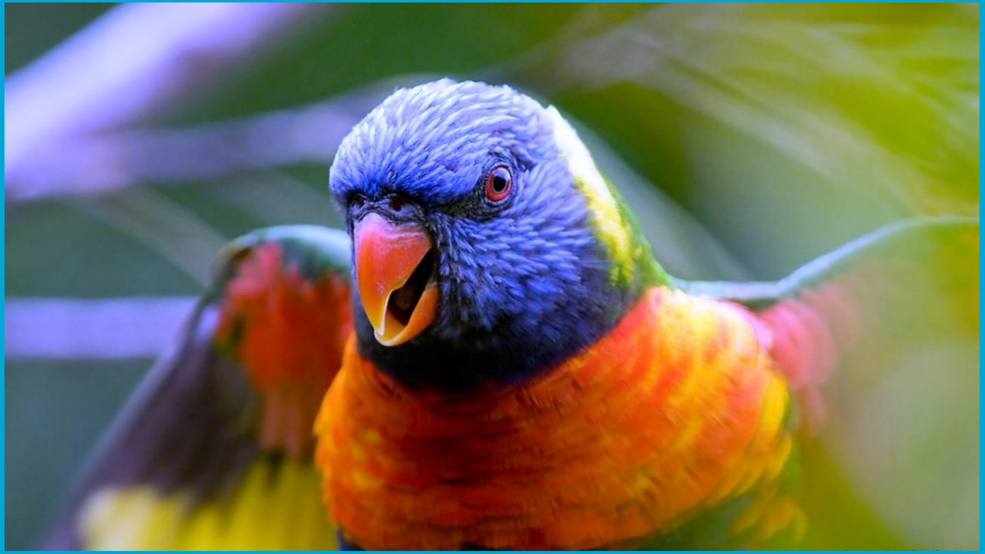
```
def image_sharp(image):
    kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
    sharpened = cv.filter2D(image, -1, kernel)
    return sharpened
```

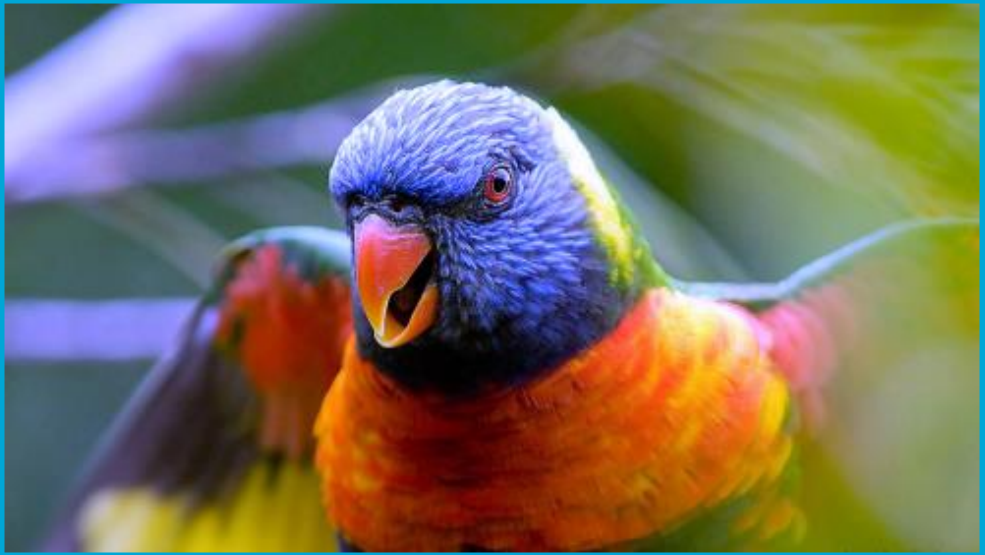
Results

Contrast Enhancement



Original Picture

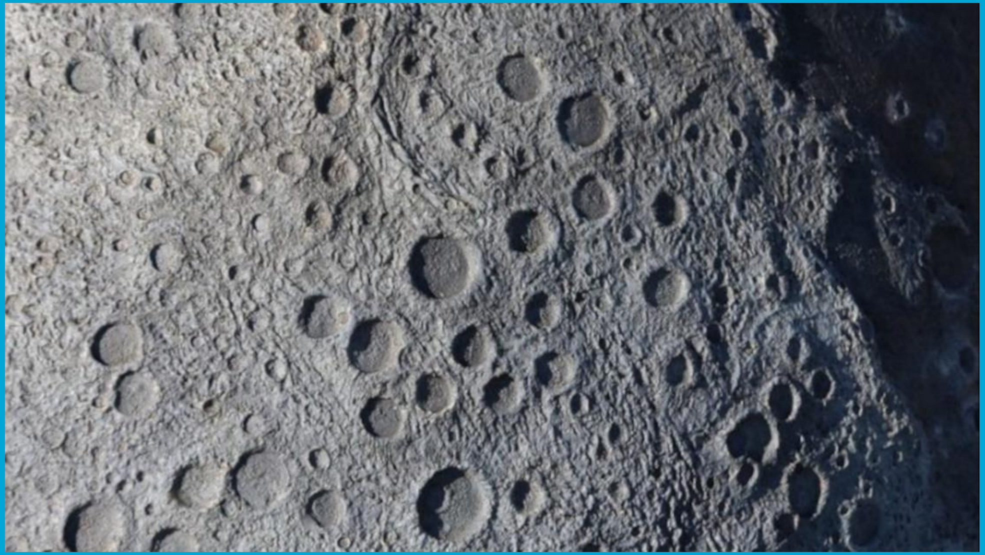




Contrast Enhancement + Sharpening

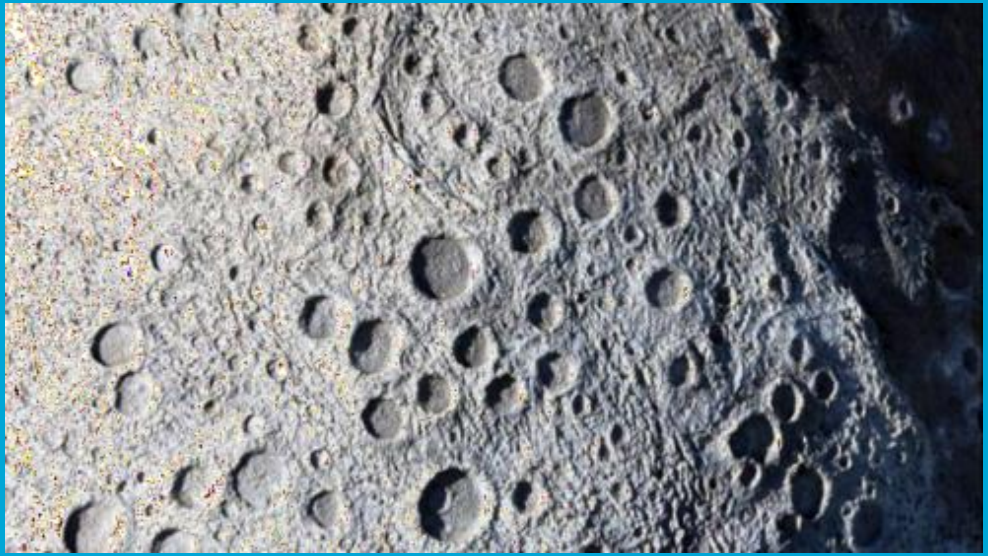
TUDelft

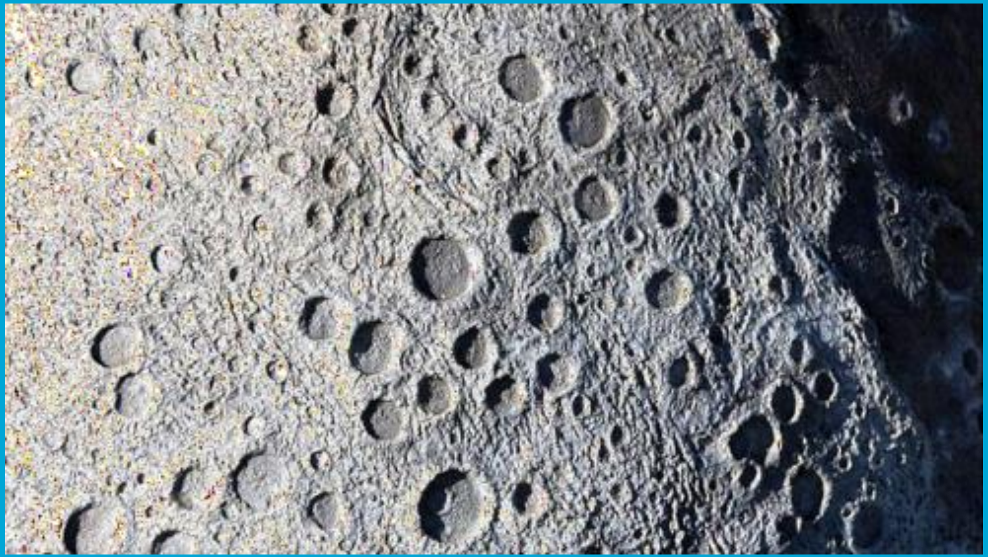
Results



Original Picture

Contrast Enhancement





Contrast Enhancement + Sharpening

TUDelft

Results

- Time measurements were recorded for each call to both the contrast and edge enhancement functions during the transmission of 60 frames.
- The performances of the final design have been evaluated in terms of (average) processing time per frame and (average) frame rate.

Module	Processing time [s/frame]	Frame Rate [frame/s]
Contrast Enhancement	0.977	1.024
Edge Enhancement	0.218	4.588
Total Pipeline	1.233	0.811

TUDelft

02

Hardware Implementation

Constraint:

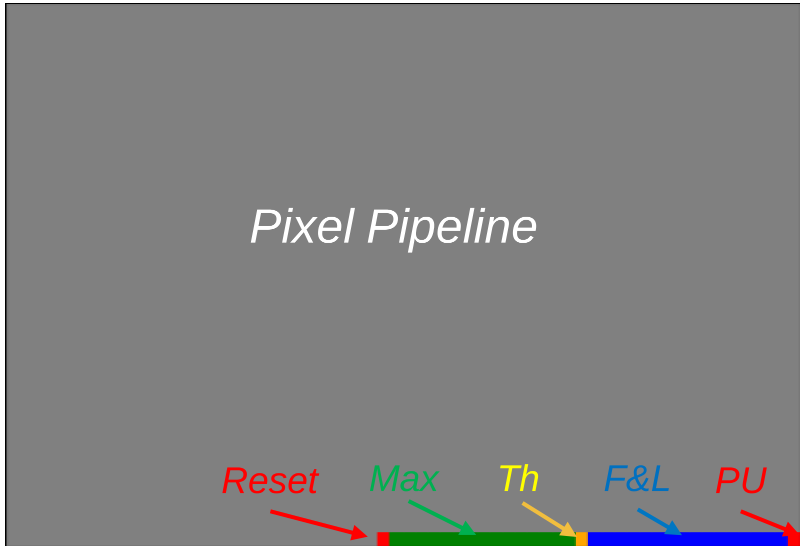
- Acquisition, processing, and output of 1 pixel per clock cycle

Limitations:

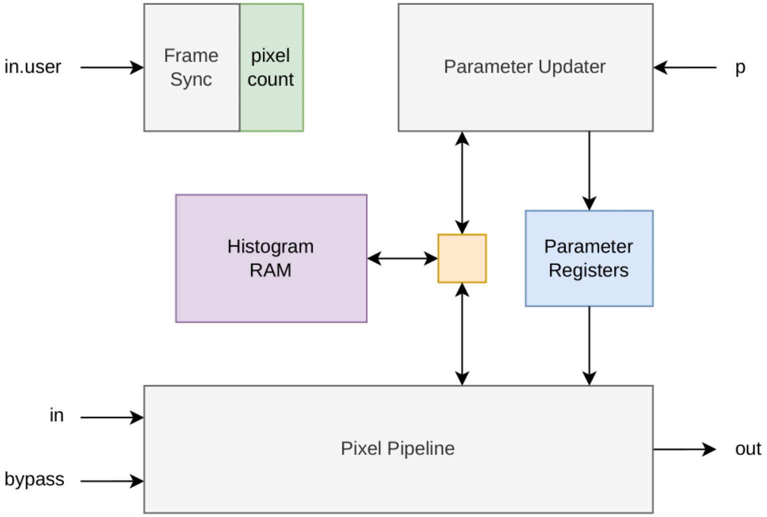
- Working on entire frames would require:
  - Too much memory
  - Too much latency

Solution:

- Stream-Processing approach
- Exploit similarities among contiguous frames



Frame structure depicting the different processing phases.

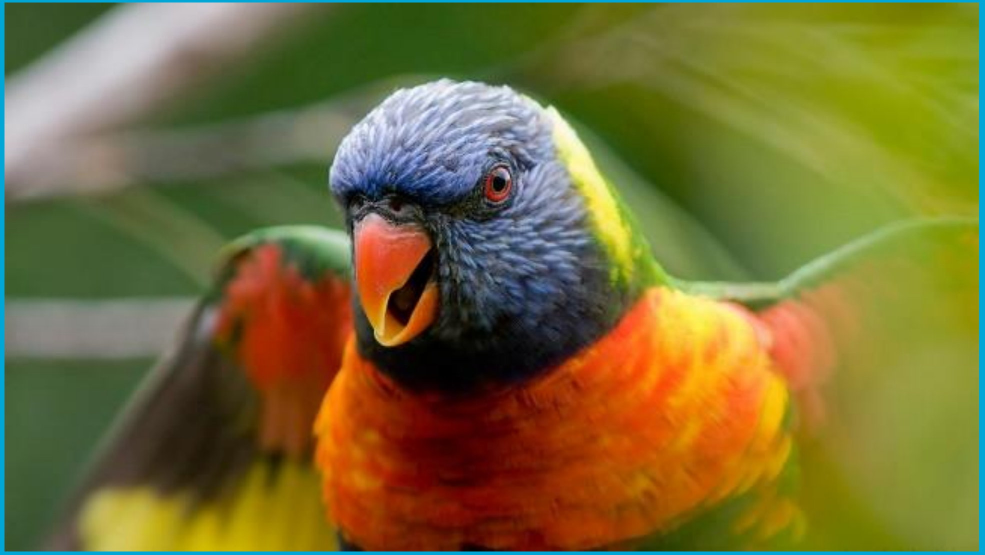


Top-level diagram of the contrast enhancement module.

Contrast Enhancement

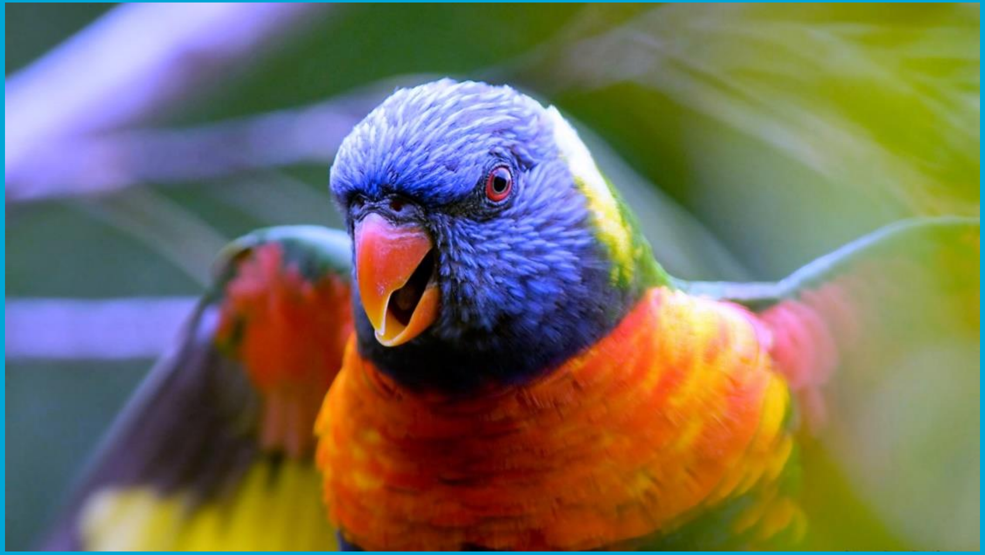
- Synchronization of computation steps by mean of a pixel counter
- Two exclusive phases:
- *Pixel Pipeline*:
  - Each received pixel increments the histogram count
  - Apply contrast enhancement according to the parameters evaluated for the previous frame
  - This phase involves 99.94% of all the frame pixels
- *Parameters Update*:
  - *Reset*: reset all the parameter concerning the previous frame
  - *Max*: find the maximum value of the histogram for each channel
  - *Th*: calculate the threshold
  - *F&L*: find *first* and *last*
  - *PU*: update all the parameters to be used for the next frame.
  - This phase involves only a portion of the last row of the frame

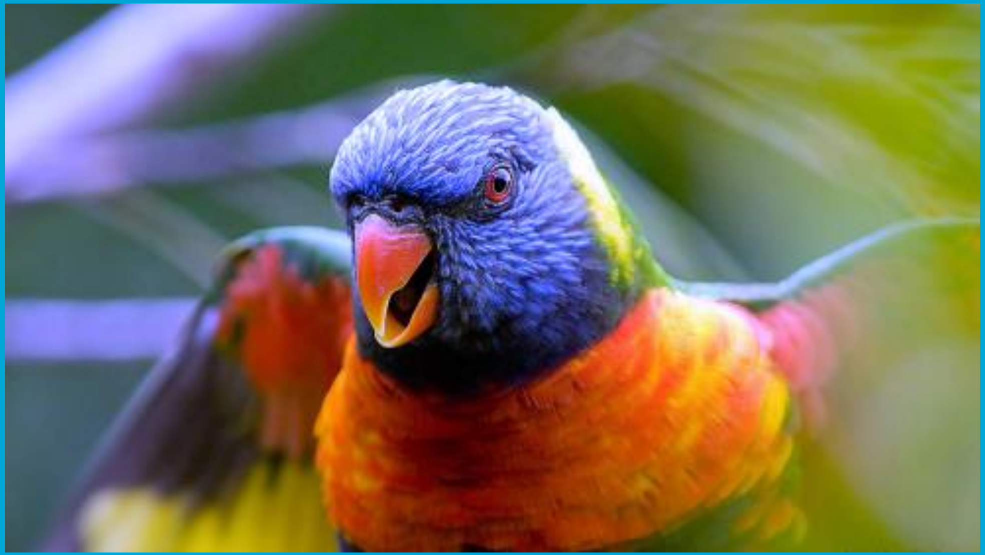
Results



Original Picture

Contrast Enhancement

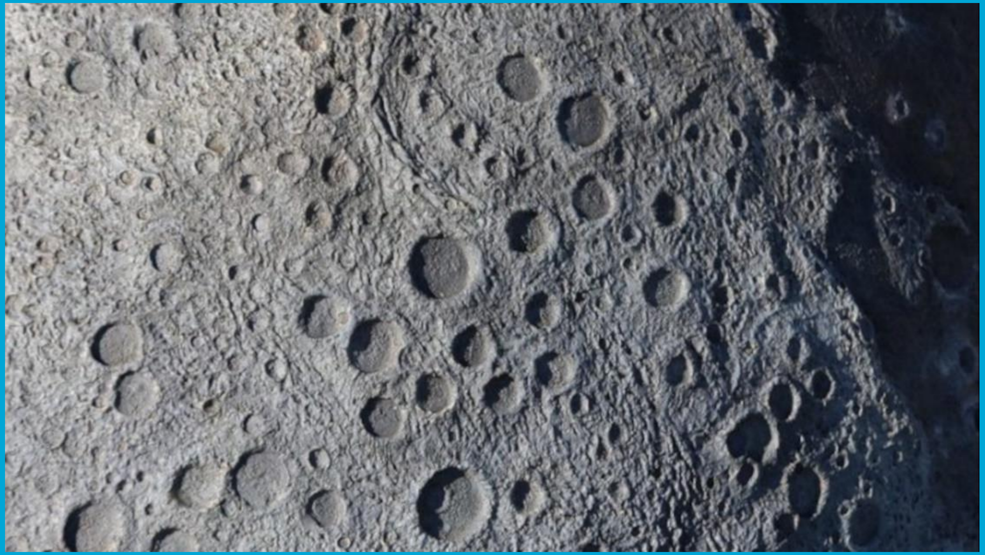




Contrast Enhancement + Sharpening

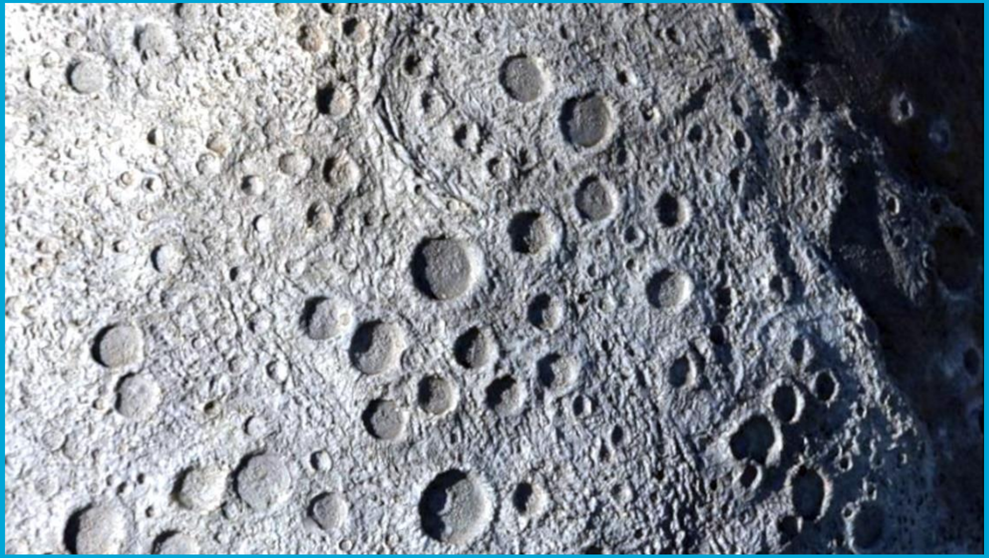
TUDelft

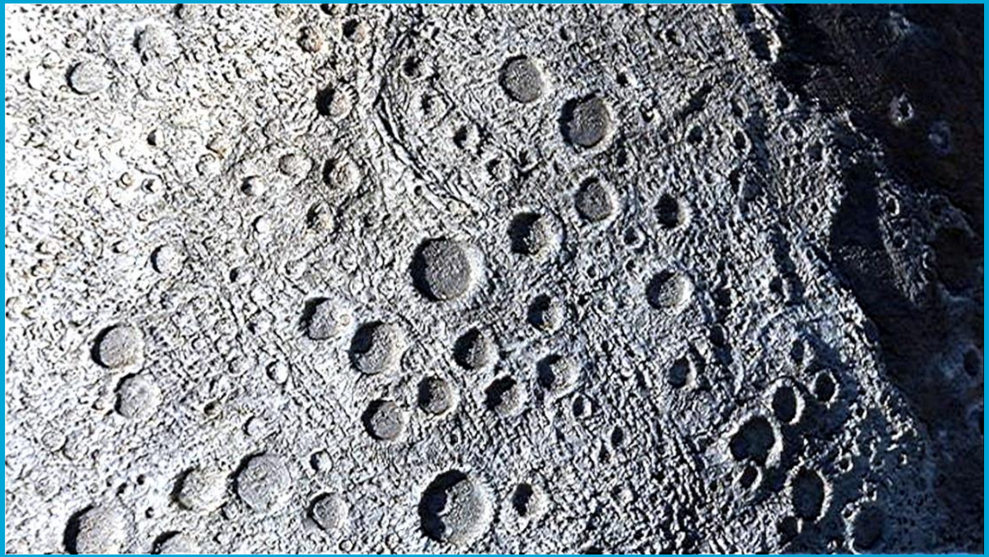
Results



Original Picture

Contrast Enhancement





Contrast Enhancement + Sharpening

Results

Processing Time:

$$7 \cdot 10^{-9} \times 921600 = 6.45\,\frac{ms}{frame}$$

Frame Rate:

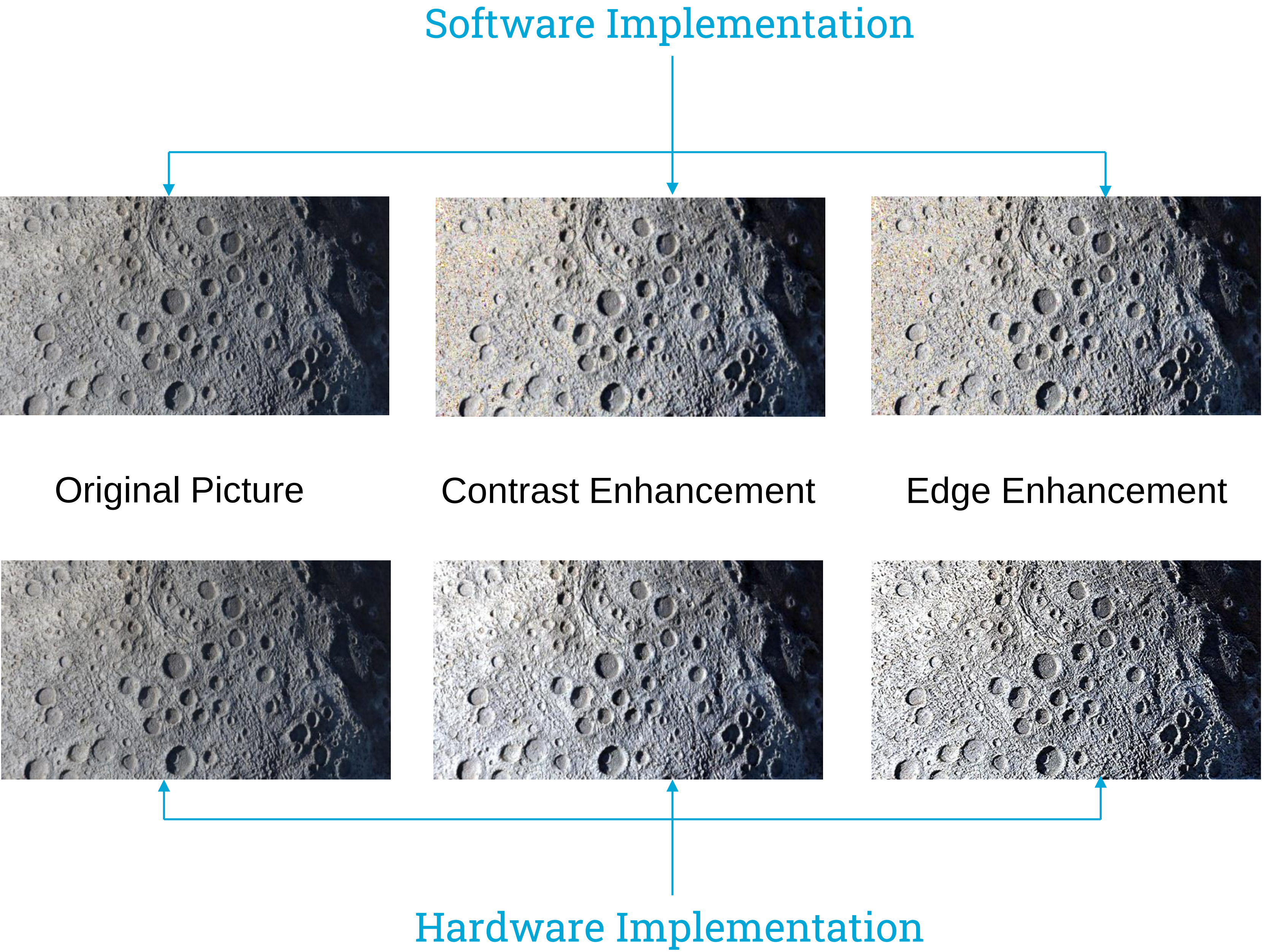
$$\frac{1}{0.00645} = 155.01\,\frac{frame}{s}$$

Latency:

$$(43 + 8) \cdot 7 \cdot 10^{-9} = 357\,ns$$

Module	Estimated Clock Period	Pipeline Stages
Contrast Enhancement	4.961 ± 1.89 ns	43
Edge Enhancement	5.081 ± 1.89 ns	8

Pixel per Frame	921600
Target Clock Period	7 ns



Software Implementation

Original Picture

Contrast Enhancement

Edge Enhancement

Hardware Implementation

Conclusion

- Hardware solution is approximately 190 times faster than the software implementation.
- Hardware solutions require much more design effort with respect to their software counterparts.
- Possible approximation errors due to the adoption of fixed point data formats in the hardware solution.

Thank you for your attention

T. Petersen, T. C. Mol, E. Cozza