Update 241221-Patch Hist

I hate drab and gloomy images, I like bright and colorful images. That was the first reason I wanted to try and use "patch hist". In this work I want to explore image manipulations that give the image the most and best colors while preserving the image content.

This is why I decided to take the route of histogram manipulation. In the area of histogram manipulation I assumed that the idea of histogram equalization is the best answer to the thought of giving the image the most colors it can have while preserving the image content.

While basic histogram equalization gives us the answer globally, I would prefer every part of the image to have our desired qualities on its own. So here we come to "patch hist" which entails preforming histogram equalization on every desired patch of the image and blending the results.

As a matter of fact I'm trying to maximize the dynamic range in each sub-patch of the image.

There was an older algorithm to try and get these results called CLAHE (Contrast-limited adaptive histogram equalization), but being dated as old as 1994, things were done then, taking into account the limits of the computing power back then, and i believe that "patch hist" will be closer to the correct answer if computing power is not a problem.

In this work I will compare basic histogram equalization, between CLAHE and my "patch hist".

**Note:** To get equal comparison, default CLAHE form openCV was used, and "patch hist" was made to work on the same patch size as CLAHE.

For presentation needs, 4 typical image categories were sampled from: selfie, full body, house and landscape.

LAB vs RGB

Because the algorithms natively work on grayscale images, and not color images, there is a need for an extension for color images like we want.

For that, I checked two main methods:

1. Transform the image from RGB-image to LAB-image, do the manipulations on the L layer of the image.   
     
     
     
   
2. Use the RGB-color image as input and work on all layers separately   
     
     
     
   

From the results it seems like the LAB manipulation looks more natural and easier for our eyes, and the RGB manipulation seems to be more artistic and colorful.

These results may match the theory of light, saying that light has 3 parameters: amplitude(strength), wavelength(color) and polarization (that normal cameras do not measure). So doing the LAB manipulation works on equalizing amplitudes as a whole, while doing the RGB manipulation works on equalizing amplitudes of specific wavelengths.

Histogramic Results

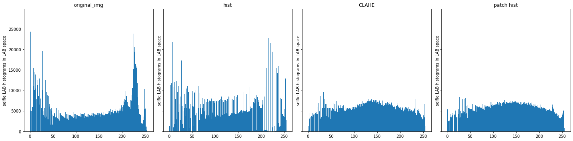
Diving into the results we see the histograms of the images that were scored for uniformity with the formula of:

Standard deviation of the histogram bins sizes,

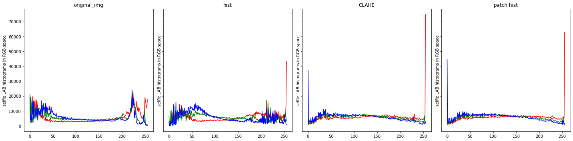
Divided by mean bin size (when the lower is the better).

**Histograms of the LAB images**

*selfie*

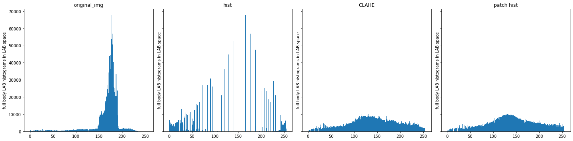


{'original\_img': [0.7058034378960452], 'hist': [0.7151122267807238], 'CLAHE': [0.24924985320994608], 'patch hist': [0.2516142507954908]}

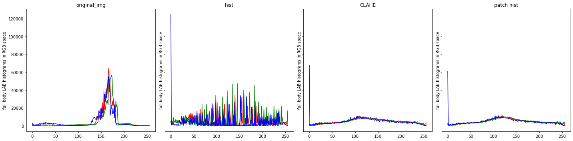


{'original\_img': [0.7086745, 0.6028355, 0.64519906], 'hist': [0.58059615, 0.39711758, 0.6103342], 'CLAHE': [0.78828806, 0.2729273, 0.4899424], 'patch hist': [0.66105014, 0.30472338, 0.55346173]}

*full body*

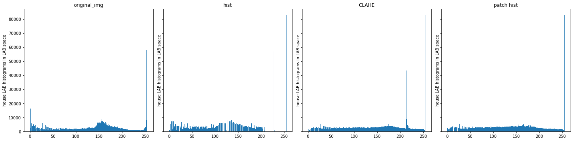


{'original\_img': [2.2946370588627882], 'hist': [2.219707850190901], 'CLAHE': [0.5186847276591188], 'patch hist': [0.5299481262156629]}

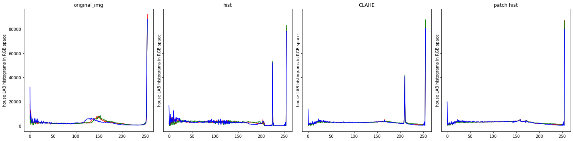


{'original\_img': [2.2625558, 2.2192402, 2.0066845], 'hist': [1.225661, 1.7677866, 1.9837741], 'CLAHE': [0.5430424, 0.5038913, 0.9523546], 'patch hist': [0.549263, 0.52038413, 0.8891966]}

*house*

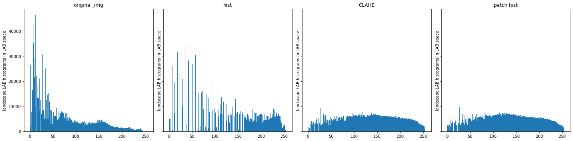


{'original\_img': [2.073258178260794], 'hist': [2.082099494290935], 'CLAHE': [1.8519052212105487], 'patch hist': [1.656989323842216]}

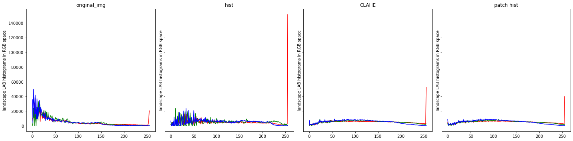


{'original\_img': [2.1905994, 2.0892348, 2.1758971], 'hist': [1.9297587, 1.9835548, 1.9394251], 'CLAHE': [1.9115356, 1.923591, 1.8072177], 'patch hist': [1.7459835, 1.7339046, 1.6494647]}

*landscape*



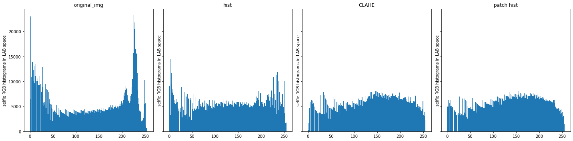
{'original\_img': [1.2757327737454527], 'hist': [1.154441256128386], 'CLAHE': [0.29849611279563265], 'patch hist': [0.312950941110327]}



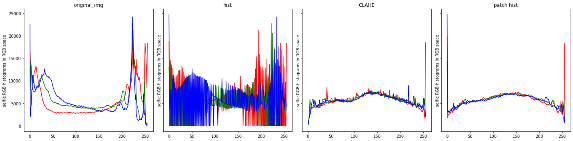
{'original\_img': [1.0880476, 1.1247098, 1.4273357], 'hist': [1.8267477, 0.67762226, 0.6771148], 'CLAHE': [0.61914515, 0.32645077, 0.48307875], 'patch hist': [0.49223813, 0.34233597, 0.49325204]}

**Histograms of the RGB images**

*selfie*

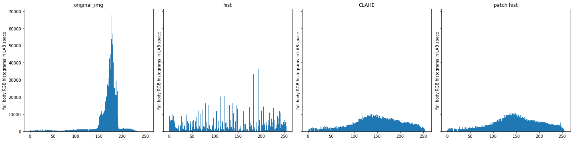


{'original\_img': [0.6558027540253926], 'hist': [0.32117349335967177], 'CLAHE': [0.2627710688891346], 'patch hist': [0.25774450812842853]}

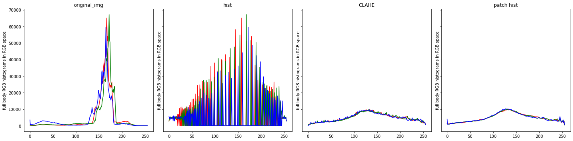


{'original\_img': [0.67381364, 0.5354483, 0.61327475], 'hist': [0.79181457, 0.6720693, 0.73950785], 'CLAHE': [0.26647118, 0.19973545, 0.19980282], 'patch hist': [0.31817862, 0.22424187, 0.28657436]}

*full body*

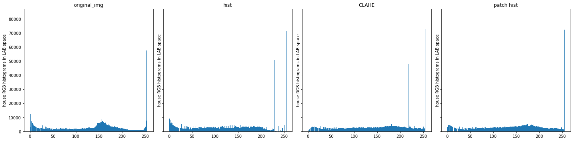


{'original\_img': [2.2847094048364376], 'hist': [1.1066256425906116], 'CLAHE': [0.5620635966273264], 'patch hist': [0.5836477215021662]}

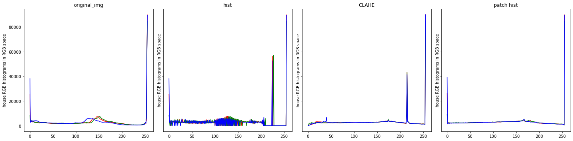


{'original\_img': [2.2173083, 2.219839, 1.9524856], 'hist': [2.2395191, 2.2445946, 1.981456], 'CLAHE': [0.512943, 0.5035514, 0.49991897], 'patch hist': [0.51934355, 0.5181008, 0.5354035]}

*house*

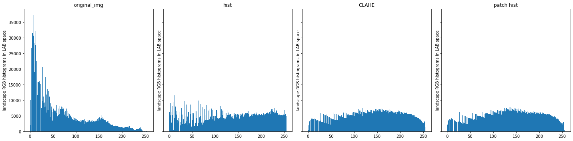


{'original\_img': [2.058046146506148], 'hist': [1.755100107288814], 'CLAHE': [1.711627246089098], 'patch hist': [1.446827327274717]}

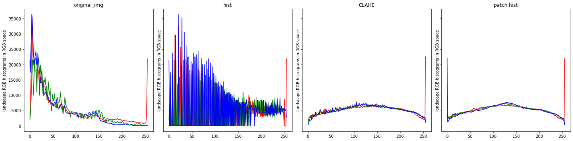


{'original\_img': [2.1807044, 2.05212, 2.210547], 'hist': [2.2188263, 2.0984893, 2.244731], 'CLAHE': [1.9774343, 1.8752652, 1.9603314], 'patch hist': [1.8498175, 1.6840404, 1.9293033]}

*landscape*



{'original\_img': [1.172096515356124], 'hist': [0.3163984379766023], 'CLAHE': [0.28061545882816147], 'patch hist': [0.29433779996163395]}



{'original\_img': [1.0597273, 1.0436568, 1.2941239], 'hist': [1.1365395, 1.1041607, 1.3336961], 'CLAHE': [0.31881455, 0.23869422, 0.26458448], 'patch hist': [0.3321455, 0.26011604, 0.34795547]}

It seems now that CLAHE generates the smoothest histograms and the first version of "patch hist" is even a little rougher than that.

Both versions do much better job then 'global histogram equalization', while (at least for me) CLAHE and "patch hist" are emphasizing most details in the images.

Expansion of “patch hist” results

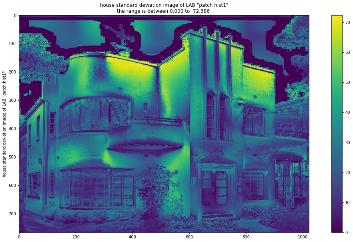
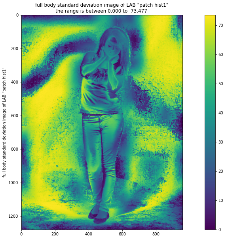
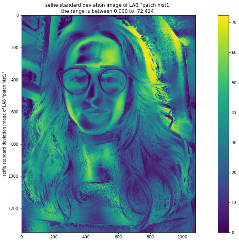
**Note:** In light of the results above, the deeper dive will focus on results from LAB images, but the same techniques can be applied to any other color space.

Because "patch hist" works on every patch in the image, it has many data points on each pixel, so we can calculate more then just the mean of the distribution.

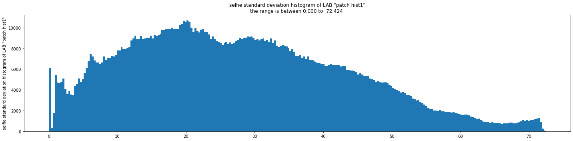
For now we will look at the standard deviation and skewness images that the deeper “patch hist” can give us.

Standard Deviation images

**Images**



**Histograms**



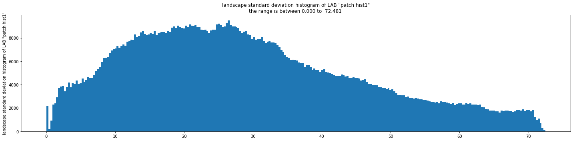
{'img\_std\_ph': [0.5501801990158296]}



{'img\_std\_ph': [0.4248094075009173]}



{'img\_std\_ph': [1.7777126195317292]}

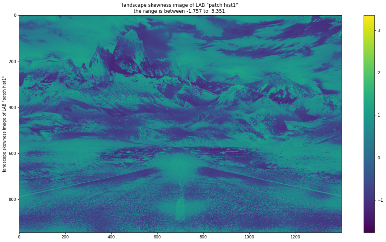
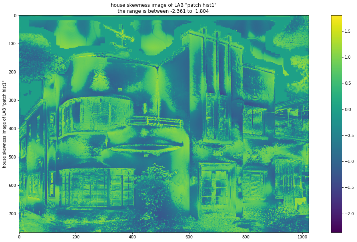
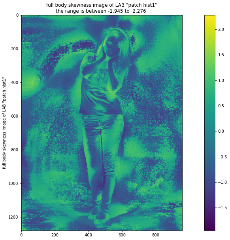
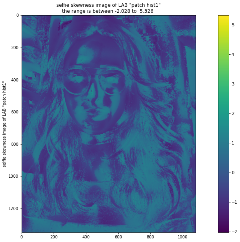


{'img\_std\_ph': [0.5120582003987958]}

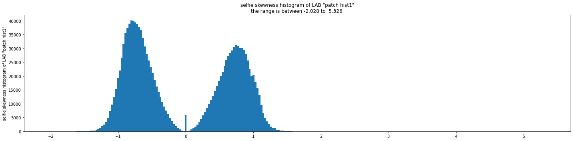
It appears that for most images the standard deviation is distributed pretty uniformly between values close to 0 up to about 70, but that shows us that the mean is not completely stable.

Skewness images

**Images**



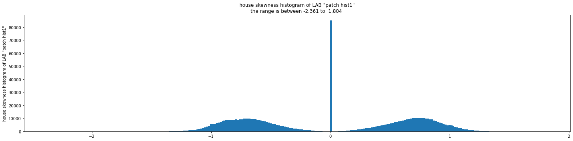
**Histograms**



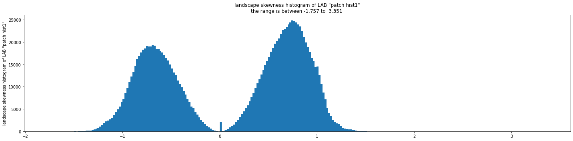
{'img\_skew\_ph': [1.8701458631368337]}



{'img\_skew\_ph': [1.238052214540793]}



{'img\_skew\_ph': [2.0240793335073093]}

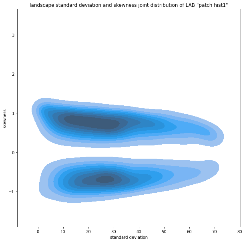
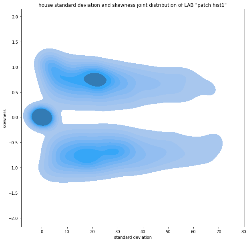
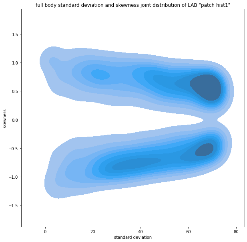
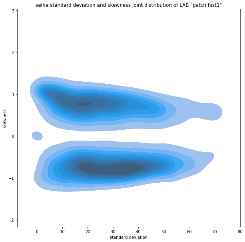


{'img\_skew\_ph': [1.4555566496417625]}

It looks like for most images the skewness is distributed in a double normal distributions with centers at around +/-0.75(mirrored at the 0 center). and this means that the distributions are mostly skewed and not normal.

Standard deviation and skewness

In light of the results above we prefer to see if these joint 'distribution of the standard deviation' and 'the skewness' show us something special or even maybe more.



skewness correction

Finally, having calculated the 'mean image' as well as the 'standard deviation image' and the 'skewness image', we can now calculate the correction of the mean 'towards the median'.

We will try a simple correction with the equation:

skew corrected image = mean - alpha \* skew image \* std image

And we will try and check results for several 'alphas'.

selfie skew corrections



full body skew correction



house skew correction



landscape skew correction



At least for my eyes, the skewness correction may give little more details' enhancement but it does not necessarily give us a better image.

**Conclusions**

It appears that CLAHE is closer to the true result that “patch hist” represent, although it can not produce 'the standard deviation and skewness images' that “patch hist” can make and that may hold some significant data.

It may be beneficiary to find the middle ground between CLAHE and “patch hist” so the work will be faster and still give us a distribution for each pixel.