COMP 478 DD - Image processing

Project - Final Report
Winter 2024
Concordia University
Presented to Dr. Yaser Esmaeili Salehani

Image restoration using IDBP

By: (Group #11)

Antoine Cantin 40211205
Ryan Li 40214839

Paper:
"Image Restoration by Iterative Denoising and Backward Projections"
By Tom Tirer and Raja Giryes

This google drive holds all docs, runnable matlab code as well as the datasets/images (there is a few Gb of data there were excluded from the moodle submission, so the full version of our submission is found in this google drive):
https://drive.google.com/drive/u/0/folders/1B4dDQso_TP_I8qZJWxpy9UYYNGxIquGY

Note some images were included to show examples of our results. However, there are too many images to include in this document. It is also harder to see details in this document due to the resizing done in the document. To clearly see full results and zoom in on local details referenced in finding,results and analysis, refer to the files "results_set12" and "results_sidd".

If one wants information about the new MatLab interface or the new code, or file structure/naming convention. All information is in the README [Appendix A] and in the "478 SIDD file structure and comparison info" [Appendix B], which are both also in the appendix of this document.

Description Of Alternative Datasets and Why Use Them

Set12 Dataset

Images located in "test_sets" folder and is downloadable at:
https://www.kaggle.com/datasets/leweihua/set12-231008

Set12 dataset is a collection of 12 grayscale images. Each image in the collection represents a change in scene or object that provides a diverse range of content for testing, experimentation, and evaluation. This diversity is needed to ensure that the IDBP algorithm doesn't explicitly favor specific types of content in images.

The 12 images are high quality clean images of size 256x256 pixels. These clean images/ground truths are commonly used as references for experimentation and evaluation. In our case, they are needed to experiment on and to measure IDBP's image denoising effectiveness with metrics PSNR and SSIM.

The dataset is a well known dataset across the image processing and computer vision communities. With how easy it is to access, it is widely used for benchmarking for evaluation and comparing performance of different image restoration algorithms by researchers and practitioners around the world.

This dataset was chosen for these reasons. We wanted a small set of images as the basis of the IDBP algorithm to ensure that it could effectively handle various types of scenes. It provides reliable performance metrics for evaluating its denoising effectiveness across different objects commonly encountered in natural images taken with a traditional camera.

Smartphone Image Denoising Dataset

Images located in "test_sets" folder and is downloadable at:
https://www.kaggle.com/datasets/rajat95gupta/smartphone-image-denoising-dataset

The Smartphone Image Denoising Dataset, SIDD for short, is a dataset of 160 paired images. Pairs consist of the same image but one is a clean version(ground truth) and the other is a noisy version. The noisy version represents images with noise that you would encounter in the real world when taking images through your phone.

What's unique about the Smartphone Image Denoising Dataset? As mentioned, these images are taken from smartphones. Typically smartphone images contain more noises due to their small sensor size and small aperture. Depending on the type of smartphone, noise can vary depending on factors such as ISO level, shutter speed, illuminant temperature, and illuminant brightness. The 160 pairs of images were taken from 12 different smartphones under these different factors. The 12 smartphones consist of the Google Pixel, the Iphone 7, the Samsung Galaxy S6 Edge, the Motorola Nexus 6, and the LG G4.

Given the new norm of owning a personal smartphone in modern times, most images are taken by smartphones nowadays. This dataset provides real world applications because it consists of various smartphone images. In addition, the different lighting conditions add to the real world application.

This dataset was chosen for the application of measuring the effectiveness of IDBP in image inpainting and denoising with photos taken from smartphones. It provides a good set of images that represents this scenario by providing ground truths and different images taken by different phones.

For this given dataset, we did not use all 160 pairs of images. 23 images were carefully picked across 4 different scenes. Not only were they chosen for their visible differences, they were specifically chosen for their differences in smartphone type, ISO level, shutter speed, illuminant temperature, and illuminant brightness. These factors were compared from both ends of their ranges. This better reflects the real world applications of image restoration of digital images.

Parameter Tuning For Both Datasets

<u>Set12 Dataset</u>

For a given dataset its scenario will be different, thus the parameter values for max iteration, sigma_e, and delta must be different. This was done to better represent the environment of how the image would be taken in the real world. In other words, the set12 dataset consists of images captured from traditional cameras rather than smartphone cameras, and therefore must be adjusted to account for this information. In the case of the set12 dataset, we have 3 scenarios.

The first scenario has values 150, 0, and 6 for max iteration, sigma_e, and delta respectively. This scenario represents a sharper image recovery due to the low sigma_e and higher delta values. Sigma_e is the standard deviation parameter of Gaussian distribution for random noise. Due to its inversely proportional to severity of noise tolerance property, a lower value produces more uncaught noise but the image restoration will have a more accurate fidelity term. The delta value will also add more weight to the fidelity term, strengthening this effect. Finer details are preserved at the expense of more noise, thus higher iteration count is required to help capture this detail.

The second scenario aims to describe a scene with less noise. However, the decrease in noise comes at the cost of losing finer details. Sigma_e is set to 12, with an offset of delta equal to 0. The max iteration is half of the first scenario at 75 iterations because less iteration is needed due to the reduced complexity of the denoising task when employing a higher sigma_e value and no additional regularization through delta.

These parameters are tuned to better interpret images, resulting in better results. As we can see, the change in values for these parameters affect the noise level and detail level of the restored image. It is about balancing these parameters to achieve your purpose by sacrificing having higher noise with finer details or having lower noise with less details. In the third scenario, we show this tuning by meeting in the middle with the first and second scenario. Max iteration is 112, sigma_e is 6, and delta is 3. This provides equal noise but also equal weight to the fidelity term, striking a balance between both previous scenarios.

Smartphone Image Denoising Dataset

In the case of the Smartphone Image Denoising Dataset, parameters should better reflect images taken through smartphones. Typically, modern smartphones take high quality images with relatively low noise. Noise levels vary depending on the type of smartphone and factors during the acquisition of the photo, such as ISO level, shutter speed, illuminant temperature, and illuminant brightness. This will be represented with 2 scenarios.

The first scenario has sigma_e equal to 0 to simulate low noise. In addition, the delta has the value of 1, indicating low noise but lower weight on fidelity term. We also adjusted the max iteration to 50 given that the image has minimal noise and does not require as much iteration to reach convergence.

In the second scenario, we’ve increased sigma_e to 5 to observe more noise. Delta is slightly raised to 2, to further enhance the denoising process while still maintaining a balance between noise reduction and preserving image details. Again, max iteration remains the same because increasing it further may lead to diminishing returns in denoising performance or significantly extend computational time without substantially improving the final result.

From README [Appendix A]
One of the changes we made to the MatLab source code is a reduction from 80% missing pixels to 50% missing pixels for the Smartphone Image Denoising Dataset*. This results in a better application and reflection of reality as digital images taken for smartphones are more likely to have less than 80% missing pixels. Even 50% is still an edge case, but for the purpose of measuring and comparing IDBP’s performance it was kept at this percentage.*

We have also downsampled Smartphone Image Denoising Dataset *images to a width of 600 pixels while dynamically adjusting height to keep aspect ratios. Most images were originally around 4000 pixels in width with a bit depth of 24. This downsampling was required to save computation time while keeping most of the finer details of the original image. A width of 600 pixels produces a good diminishing return in denoising performance and computational time.*

Note that the Smartphone Image Denoising Dataset *are colored images.IDBP works with grayscale images. To correct this, we converted the image to grayscale before inpainting and denoising.*

SIDD Samples benchmarks and results analysis

(I.) Data:

Specific images to evaluate different parameters in image acquisition and processing [Appendix B]

<scene-instance-number>_<scene_number>_<smartphone-code>_<ISO-level>_<shutter-speed>_<illuminant-temperature>_<illuminant-brightness-code>

(II.) Setup and processing:

1. Images downsampled and dimensionality-reduced: ~5328x~3000 24bit depth to 600xH 8bit depth. (~10-40Mb to ~50-100Kb)

2. Images converted from sRGB to Greyscale-256

(III.) Algorithm:

Denoiser: BM3D

IDBP initialization restoration filter: Median nan-reduced inpainting

Metrics: PSNR, SSIM

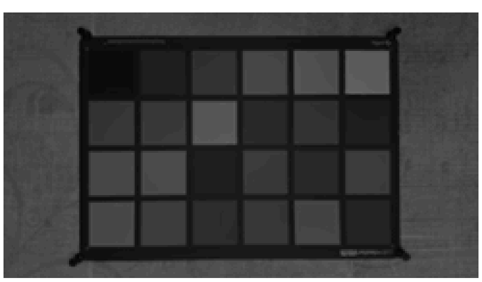
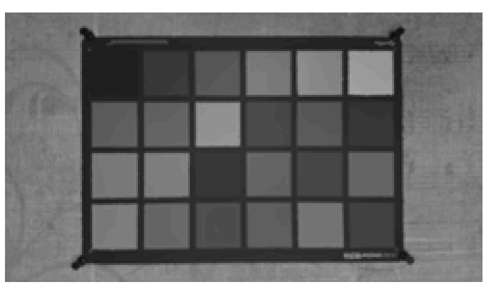
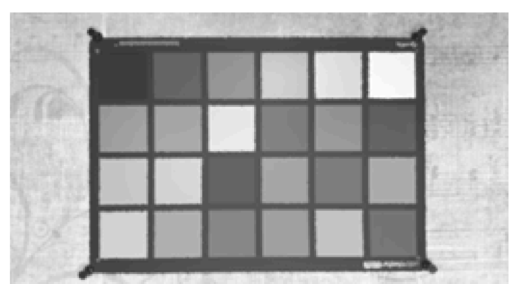
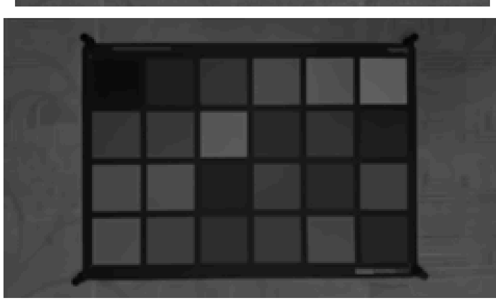
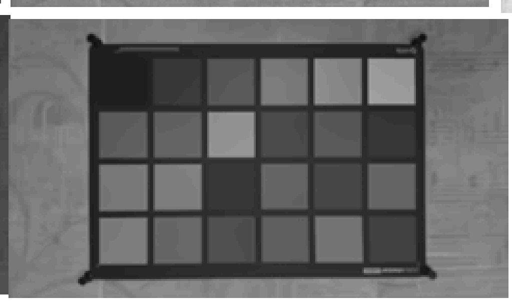
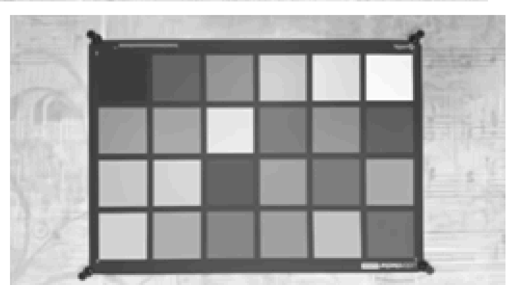
(IV.) Runs (2 Scenarios with same # iterations):

Scenario1 : Higher noise tolerance, results have more artifacts but have better sharpness and clarity. Usually closer to on a pixel by pixel basis to ground truth images and priors, but may have some small arbitrary noise details present.

Scenario2: Lower noise tolerance and more strict inpainting, results have barely any artifacts and objects contours/definitions are clear/separate, but overall sharpness and clarity is reduced; images appears more blurred. Usually closer object by object to priors.

(V.) Results:

Brightness/Exposure levels:

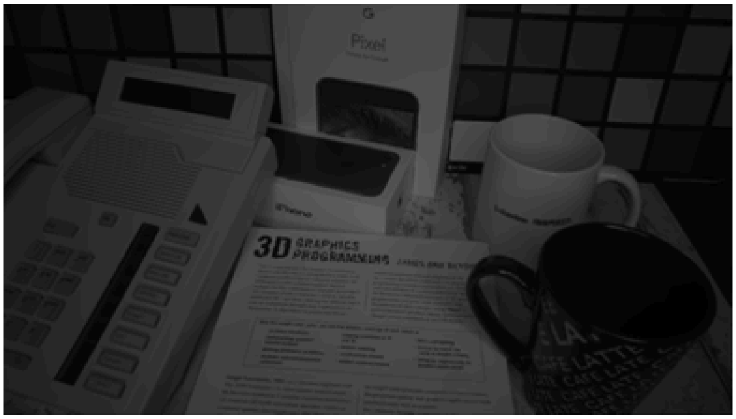
Evaluation (in terms of metrics and result usability) of Exposure/Brightness effect:

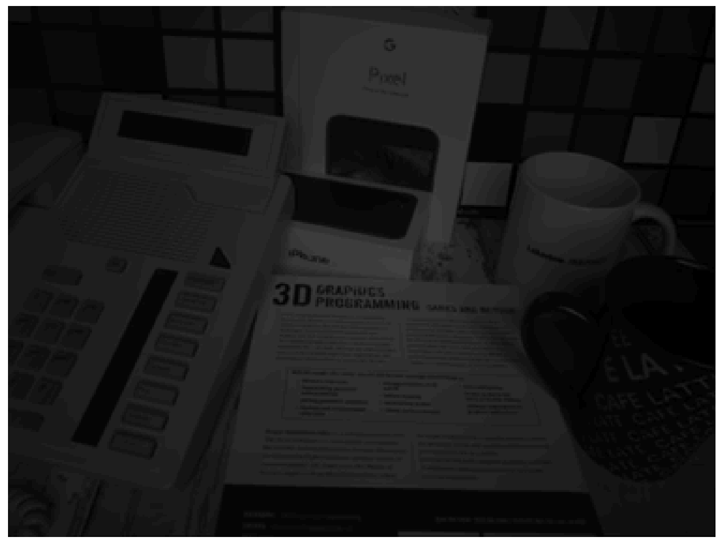
Lower exposures seem to be most beneficial on average for Scenario 1 images, but for Scenario 2 higher exposures seem to be most beneficial. These are interesting results because this means that we can improve results in most applications of common modern objects images by choosing the 3 IDBP parameters in function of our goal (sharpness or artifacts-minimalization) and our image acquisition brightness (we can use Gamma correction after to standardize images)

Illuminant temperature / saturation effect:

Results are often similar, since we convert our images to standardized greyscale so the image acquisition focusing/polarizing/emphasizing certain wavelengths of the light color spectrum doesn’t affect our result much. But for images like Scene 7 (Scrabble board) which has a lot of text and connected objects (tiles), it seems that surprisingly after inpainting IDBP, the restoration is better on average with lower/middle image temperature like 3200K-4400K (Furthermore, 5500K/high temperatures seem to have zero-padding dark border effects a little bit, which is another downside). That is likely because to restore such elements a balanced contrast is optimal and when we convert to grayscale the similar emphasis on both warm and cold colors shows up as an equalized and efficient contrast level. Since the Scene 7 also emphasized on the scrabble tiles and texts that are high-contrasting, this also made an edge-case where Scenario 2 had better clarity due to the fact that it focuses better on tiles contours and text outlines whereas Scenario 1 had a lot of texture related artifacts.

Image acquisition devices/smartphones:





Evaluation of phone (Image acquisition and encoding/compression):

Overall, LG G4 and GP Google pixels were the better phones on average with the various lighting and capture settings/conditions. Namely, GP’s pictures have excellent HDR+ properties and it shows because since our images are greyscale the dynamic range and contrast in image colors are directly shown in how equalized are our downsampled and greyscraled images. GP is also excellent in any lighting condition and namely performs very well in low temperatures and low brightness images i.e. it reproduces the brightness of the image prior accurately while preserving high clarity and pixel definition. G4 does better in regular lighting conditions but captures very well vectorized elements such fine shape contours and various different fonts. It is very good in text restoration after IDBP inpainting, even with the more blurred Scenario 2 texts are usually readable (which is rarely the case with other phones) ! These are purely empirical observations and more data could be observed if we would run more images from the datasets, more scenarios, different combination of image acquisition parameters and without downsampling, etc…; certain phones may perform better with specific pre-processing and configurations of exposure, ISO-level, shutter speed, etc… But the SIDD dataset has been known and standardized in the industry, so even if IP, S6 and N6 may perform slightly worse in some of our cases and need slightly higher error margins, their results are still overly satisfying and assuredly usable in our measurements and analysis.

Using older/lesser smartphones also allows us to test IDBP on various image priors that, as in real life applications, may have less starting “information” i.e. lesser quality / scene reproduction accuracy.

It is also worth noting that certain phones may perform better without downsampling as depending on the image aspects, oversampling or downsampling may similar effects as something like digital zoom or digital focus may have on images. Digital pre-processing (which is never truly lossless) like these are avoided in image restoration because it affects fidelity of priors since IDBP being itself is a digital/artificial operation and measurements/application are made harder if there are already digitization effects in the priors. It is better to use ground-truth images optically captured to avoid starting from priors that already have)

ISO level, shutter speed and lens settings:

Shutter speed isn't usually too much of an issue as it more used to quantify motion blur and frame resolutions when pictures are taken of objects in motion (which is not our case here apart from possible minor tripods/person holding the phone misadjustments). ISO-levels on the other hand has some effects. On most scenarios, parameters and scenes, as with Scene 3 (100x iso levels we can see creates over-fitted / over-focus light spots near the bottom of the image where the angle if the illuminant presumably as mostly projected onto); we can see that it affects mostly the capture illumination intensity distribution. While it does affect result clarity a little bit, it doesn't usually affect our IDBP inpainting algorithm. Maybe on local iterations it may increate uncertainties, but with high amounts of iterations the overall result in terms of metrics and perception were still satisfying in our tested benchmarks.

If we look at the Scenario 1 of Scene 9 (the textbooks, 2 pictures above) in normal brightness and compare the images with ISO-level 100 vs 1600, we can see that because the illumination temperature made dark-light capture a slightly more challenging issue in terms of contrast, that actually a higher ISO-level of just 16x here (vs 100x in Scene 3) actually had an effect on sharpness and clarity ! The higher ISO-level picture being exposed longer and capturing more light made it darker (thus more accurate) but also more close/detailed relative to the image prior and ground-truth (And it also removed a lot of artifacts that were usually there in most Scenario 1 images and scenes). This is an interesting result as it shows that lighting, color temperature (even if we use grayscale) and brightness can decide if ISO-level has a significant effect at all on the image restoration inpainting and denoising processes depending on the objects and scenario. The easy parameter tuning offered by IDBP makes it a very good algorithm to use here as we can regularize our algorithm parameters while benchmarking the above-mentioned image acquisition parameters in order to obtain optimal results in the scope of real life application.

Full submission link [REMINDER]:

https://drive.google.com/drive/u/0/folders/1B4dDQso_TP_I8qZJWxpy9UYYNGxIquGY

ROLES AND TASKS BY TEAM MEMBERS

Phase 1:

Ryan Li - Role: Team member

- Create code based on research paper and made modification.
- Write introduction and purpose
- Tested and benchmarked various denoisers, models and IDBP setup filters.
- General reviews of the IDBP paper; deblurring,inpainting, generic denoising

Antoine Cantin - Role: Team member

- Paper reviews and python notebooks + READMEs
- Mathematical summaries, elaborations and overviews (proofs and expansion of mathematical concepts/utilities from the paper)
- Index terms explanation + theoretical elaborations, conceptual analysis and application and testing into actual algorithms
- Graphing, analysis, optimization of running benchmarks/running IDBP

Phase 2:

Ryan Li - Role: Team member

- Problem statement
- Inpainting algorithm
- Metrics (PSNR, SSIM, presentation and analysis of resulting benchmark graphs for different scenarios)
- Results and examples

Antoine Cantin - Role: Team member

- IDBP conceptual and mathematical first slides, equations (iterative denoising slides, backward projections slides explanatory)
- Inpainting models, problems , general examples that serve purpose/motive of presentation
- Parameter tuning and setup of our benchmarks (sigma_e, delta,k_max) and how they change in our scenarios #1 and #2
- New dataset researched and influence of image priors, mix of dataset, pragmatic part planned for Phase 3

Phase 3:

Ryan Li - Role: Team member

- More research into datasets
- Description of dataset + motives for final phase
- Why use dataset + choice of SIDD dataset, early and formal tests/benchmark runs
- Choosing images from dataset
- Parameter tuning for datasets, test for different scenarios and choices of parameters based on graphs and results from previous phases
- Modification of MatLab code to work on new dataset + code refactoring

Antoine Cantin - Role: Team member

- Matlab Command line interface to match and filter out images in SIDD dataset + matlab interface viewer to view its SIDD parameters/properties as well as resulting matched images
- READMEs and formal code doc artifacts, formatting of documentation + organization of submission
- Benchmark, parameters and results full analysis based on different settings and SIDD parameters
- Conceptual elaboration and main findings from datasets, possible future and present applications of our findings
- Concepts and theoretical completion of past mentioned topics in previous phases

Appendix A [Phase 2&3 Part of the README.md]

> **NB: In code documents and README as below, “Phase 2” refers to Phase 2 AND Phase 3. It means the second/final part of the code**

Project Phase 2, and Phase 1 IDBP Antoine Cantin 40211205 and Ryan Li 40214839

Phase 2 addition#1: IDBP changes
One of the changes we made to the MatLab source code is a reduction from 80% missing pixels to 50% missing pixels for the Smartphone Image Denoising Dataset. This results in a better application and reflection of reality as digital images taken for smartphones are more likely to have less than 80% missing pixels. Even 50% is still an edge case, but for the purpose of measuring and comparing IDBP's performance it was kept at this percentage.
We have also downsampled Smartphone Image Denoising Dataset images to a width of 600 pixels while dynamically adjusting height to keep aspect ratios. Most images were originally around 4000 pixels in width with a bit depth of 24. This downsampling was required to save computation time while keeping most of the finer details of the original image. A width of 600 pixels produces a good diminishing return in denoising performance and computational time.
Note that the Smartphone Image Denoising Dataset are colored images.IDBP works with grayscale images. To correct this, we converted the image to grayscale before inpainting and denoising.

For more information, Read the Final Report document ! It has all the information about parameters, code, datasetss and results/findings !

Phase 2 addition#2: Matlab SIDD parameters interface

Make Sure to add all paths (folders and subfolders) in both matlab online and desktop for the paths to find the necessary images and scripts

VIDEO DEMO OF THE MATLAB INTERFACE CODE
<video controls src="DEMO_of_SIDD_CLI_interface2.mp4" title="DEMO OF THE MATLAB INTERFACE CODE"></video>

Description
For our phase 2, we have used specifically selected samples of the SIDD dataset to test multiple image acquisition conditions, parameters, settings and setups.
This tool presents a very useful data filtering, viewing and sampling command-line interface that allows users to enter filters to find images in the SIDD dataset that matches their needs for analysis and processing; and it presents them as well as their properties in a matlab interface.

Interface

CLI to enter parameters
- scene_number
- smartphone_type
- iso_level
- shutter_speed
- illuminant_temperature
- brightness

These are the parameters that can be filtered. The CLI will display cells with various values that repesent all the possible values that a user can enter for the specific parameter (one of the above).
user can enter 4 different inputs:
- List e.g. 432,12,2,1,6 or G4,N6,GP
- range e.g. 1:8, 2:2:10
- single value e.g. 234 or G4
- Just press enter, enter no value to ignore and not filter for this parameter e.g. for smartphone_type, that means match all images taken by any smartphone and that fit the other parameters.
The viewer should launch once inputs/skips for each parameters are entered. It should print paths in the

```
command line console to allow user to directly copy paste the path of images and paste them manually in our
IDBP driver to test the algorithm for inpainting on it.


### Viewer
3 buttons in the viewer:
- Prev image: See the previous matched image
- Next image: See the next matched image
- Show parameters: Show info about image current displayed


## Code files

- SIDD_CLI_interface2.m
- ImageData.m : dataclass that represents an image and its specified SIDD properties
- ImageDataset.m : dataclass that represents a datastructure collection ImageData objects and that provdes
methods to use and filter certain ImageData based on parameters
- test_sets/SIDD_Small_sRGB-dataset/Data: is where data should be sorted in folders with each 2 images in
them (Ground truths and noisy)
```

REST OF README IS IN README.MD

Appendix B [478 SIDD file structure and comparison info.pdf]

SIDD Image naming convention:

<scene-instance-number>_<scene_number>_<smartphone-code>_<ISO-level>_<shutter-speed>_<illuminant-temperature>_<illuminant-brightness-code>.PNG

Scene-instance-number - Unique Identifier

Scene Number - Same Scene With Different Factors

Smartphone Code

- GP: Google Pixel
- IP: iPhone 7
- S6: Samsung Galaxy S6 Edge
- N6: Motorola Nexus 6
- G4: LG G4

ISO - Range: 100-3200

Shutter Speed - Range: 20-4000

Illuminant Temperature - Range: 3200-5500

Illuminant Brightness Code

- Low Light
- Normal Light
- High Exposure

Image Names For Comparison of Different Factors

<u>Scene 1</u> - 8 Unique Images Total

<u>Smartphone Code</u>
0002_001_S6_00100_00020_3200_N
0017_001_GP_00100_00060_5500_N
0022_001_N6_00100_00060_5500_N
0033_001_IP_00100_00160_3200_N

<u>Illuminant Brightness Code</u>
0001_001_S6_00100_00060_3200_L
0002_001_S6_00100_00020_3200_N
0003_001_S6_00100_00060_3200_H

<u>ISO</u>
0014_001_S6_03200_01250_3200_N
0002_001_S6_00100_00020_3200_N

<u>Shutter Speed</u>
0030_001_IP_01600_02000_5500_N
0002_001_S6_00100_00020_3200_N

<u>Illuminant Temperature</u>
0017_001_GP_00100_00060_5500_N
0002_001_S6_00100_00020_3200_N

<u>Scene 7</u> - 5 Unique Images Total

<u>Smartphone Code</u>
0142_007_N6_00100_00100_4400_N
0155_007_GP_00100_00100_5500_N
0163_007_IP_00100_00100_3200_N

<u>Illuminant Brightness Code</u>
0147_007_G4_00100_00100_4400_L
0142_007_N6_00100_00100_4400_N

<u>ISO and Shutter Speed</u>
0142_007_N6_00100_00100_4400_N
0145_007_N6_03200_03200_4400_N

<u>Illuminant Temperature</u>
0155_007_GP_00100_00100_5500_N
0163_007_IP_00100_00100_3200_N

<u>Scene 9</u> - 3 Unique Images Total

<u>Smartphone Code</u>
0194_009_IP_01600_04000_3200_N

<u>Illuminant Brightness Code</u>
0194_009_IP_01600_04000_3200_N
0195_009_IP_01600_04000_5500_L

<u>ISO and Shutter Speed</u>
0192_009_IP_00100_00200_3200_N
0194_009_IP_01600_04000_3200_N

<u>Illuminant Temperature</u>
0194_009_IP_01600_04000_3200_N
0195_009_IP_01600_04000_5500_L

<u>Scene 3</u> - 7 Unique Images Total

<u>Smartphone Code</u>
0057_003_G4_00100_00125_5500_L
0060_003_S6_00100_00100_4400_L
0066_003_GP_00100_00200_3200_L
0073_003_IP_00200_01000_5500_L

<u>Illuminant Brightness Code</u>
0057_003_G4_00100_00125_5500_L
0054_003_N6_00100_00160_5500_N

<u>ISO and Shutter Speed</u>
0063_003_GP_00100_00100_4400_N
0065_003_GP_10000_08460_4400_N

<u>Illuminant Temperature</u>
0066_003_GP_00100_00200_3200_L
0057_003_G4_00100_00125_5500_L