

**Department of Electrical Engineering**

Project Name:

Image Classification in Low Light

Student name: Rom Hirsch and Yarom Swisa

ID: 313288763 and 203675814

Mentor: Dr. Sasha Apartsin

Date of Submission: 31/05/2022

|  |  |
| --- | --- |
|  | Mentor approval:  *Approved 31/05/2022*  *Sasha Apartsin* |

**Table of Contents**

[**1. Introduction**](#_3znysh7) **5**

[**2. Project Objectives**](#_u6cbq7spx9wz) **7**

[**3. Success Metrics**](#_bu6ccj13vqem) **7**

[3.1. Photo-realistic algorithm](#_17t1vs4co47d) 7

[3.2. Classification measures](#_m0w2rdn2d5zh) 7

[3.2.1. Classification Accuracy](#_ogr5aj5nk4jl) 7

[3.2.2. Precision](#_3dy6vkm) 8

[3.2.3. Recall](#_1lg1o9rrwwuf) 8

[3.2.4. F1-score](#_lf6gmigrpwz4) 8

[**4. Literature review**](#_n0368o98l7zf) **8**

[4.1. Low-Light Image Enhancement Methods](#_1t3h5sf) 8

[4.2. Deep learning classification methods](#_4d34og8) 9

[**5. Methods**](#_6pry3dezzut2) **10**

[5.1. Block Diagram(Experiment and Implementation workflow)](#_17dp8vu) 10

[5.2. Description of Project Phases](#_izmxbsro3kbv) 10

[**6. Engineering challenges**](#_3rdcrjn) **12**

[**7. Division of work between the partners:**](#_26in1rg) **12**

[**8. Required tools**](#_19w8i07lgeyp) **13**

[8.1. Programs language](#_nnc0mqdeqaqk) 13

[8.2. Develop environments and data analysis:](#_h4p0ufa4dk1b) 13

[**9. Expected outcomes/deliverables**](#_jqag8iqg6co4) **14**

[**10. Work plan**](#_1ksv4uv) **15**

[**11. References**](#_pgmld5g5xnuz) **16**

Abstract

Accurate image classification is a key requirement in many computer vision systems. One of the most critical factors for object classification accuracy is illumination conditions under which the images are acquired from sensors. In many practical applications including automotive, video surveillance and photography, object classification for low light images is required.

Object classification becomes more challenging in low light images due to various image degradation resulting from low light acquisition conditions: color bias, unknown noise, detail loss and halo artifacts. The conventional deep learning models frequently underperform on low light images since low light images have different properties compared to the images that were used during the model training phase.

The objective of the project is to evaluate different methods for improving the classification accuracy in low light images including image enhancement prior to classification and training Deep Learning models with low light images synthetically generated from pristine image dataset.

# 

# 1. Introduction

Image classification is a main element in many computer vision systems. Robust image classification algorithms are important in many areas, such as traffic monitoring, collision avoidance, face recognition, etc.

**Classification** is a process of assigning a class to the context of an image. Some of the main classification methods based on Machine learning algorithms include Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and logistic Regression. The biggest advantage of these classifiers is their ability to perform classification by using relatively small datasets. However, for larger datasets and complex problems they are limited. In order to deal with large datasets and more complex problems there are classification methods based on Neural Network (NN), NN provides good performance while working with large datasets.

**Neural Network (NN)** is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. NN rely on training data to learn and improve their accuracy over time. NN architecture includes an input layer, hidden layer, and an output layer (figure 1).

A picture containing timeline

Description automatically generated

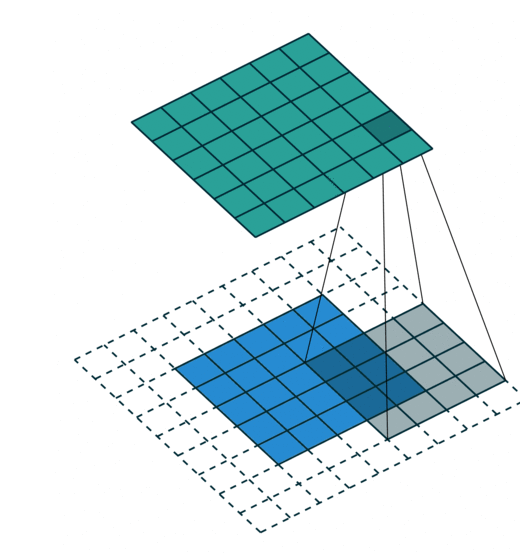
*Figure 1 – NN architecture example*

In case of **image classification,** the input dimension usually defined as Width\*Height x 3 for RGB and Width\*Height x 1 for gray scale format.Hence the number of calculations that NN should perform in order to provide classification depends strictly on the image size. Extensive calculations are the main factor that limits the possibilities of NN to perform classification and detection on large size images. The most common methods for image classification are based on Convolutional Neural Network (CNN) architecture.

**Convolutional Neural Network (CNN)** is using a convolution in place of general matrix multiplication in their layers (figure 2). They are specifically designed to process pixel data and are used in image classification. Image classification with CNN works by sliding a kernel across the input image to capture relevant details in the form of features.

One of the problems image classifications need to deal with is Low-light images which we encounter in many modern systems, such as surveillance, and autonomous driving etc. Unfortunately, low light images Classification is a challenging task, low-light conditions are not only low in brightness, but they also suffer from many other problems such as color bias, unknown noise, detail loss and halo artifacts.

*Figure 2- convolution operation*



There are some methods to overcome this problem. The first is to enhance the low light images before passing through the classifier model and the second is to train the classification NN model on a data set that includes low light images in the first place.

In this project we will use advanced low-light images enhancement methods and evaluate their contribution to the classification accuracy. The results will be compared with a model that was trained from the beginning for low-light images. In our project we will be focused on the goal of low-light images classification accuracy improvement. Our main research questions are as follows:

**RQ1:** What low-light image enhancement methods can be used to improve classification accuracy of deep learning models?

**RQ2:** Does including low-light images in the training set will improve classification accuracy of deep learning models?

**RQ3:** Which is the best method for improving the classification accuracy of deep learning models?

# 

# 2. Project Objectives

The objective of this project is to implement and compare two different approaches of dealing with low light image classification:

* **Approach A**: Apply image enhancement before using the conventional image classification model.
* **Approach B**: Train the image classification model using synthetic low light images as part of the training dataset

We will compare the results of the different approaches and evaluate the best solution. For achieving the above adjective, the following goals must be met

* **G1**: Design, implement and validate a photo-realistic algorithm for generation of synthetic low light images.
* **G2**: Research, select and implement existing image enhancement algorithms for improving low light images
* **G3**: Design and train a new image classification model with the help of synthetic low light images
* **G4**: Design and execute the performance analysis and comparison study

# 3. Success Metrics

## 3.1. Photo-realistic algorithm

In order to evaluate the algorithm, we will collect real low light images. We will analyze the illumination distribution of real low-light images and check the illumination distribution of the synthetic fitting to the real low light images.

## 3.2. Classification measures

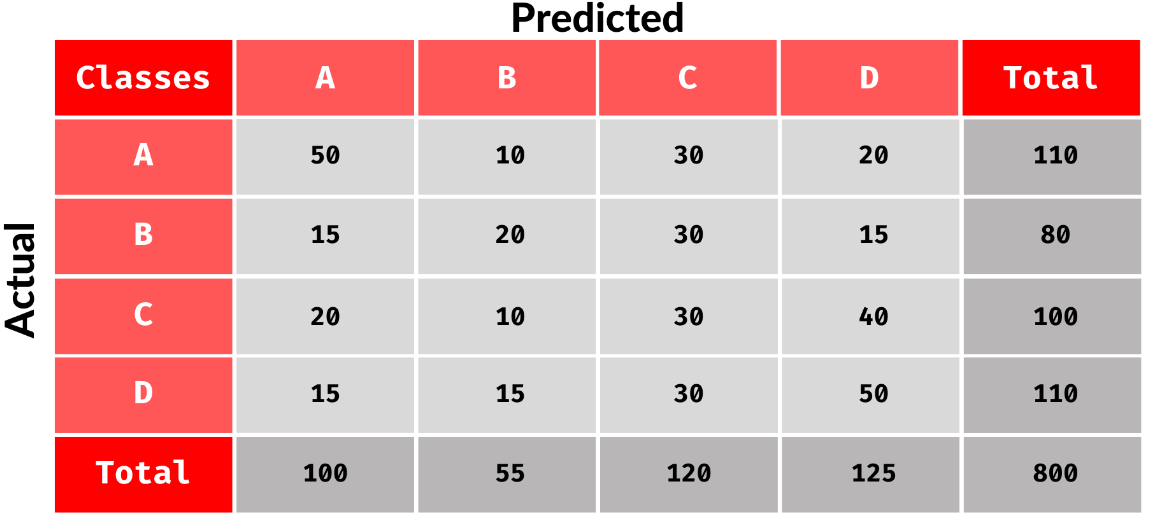
The different methods for classification will be measured by 4 measures:

* accuracy
* Precision
* Recall
* F1-score

These measures will guide us to conclude the best approach for low light image classification.

### 3.2.1. Classification Accuracy

Classification accuracy can be calculated from a confusion matrix. A confusion matrix is a tabular way of visualizing the performance of your prediction model. Each entry in a confusion matrix denotes the number of predictions made by the model where it classified the classes correctly or incorrectly.



The accuracy is defined as Accuracy = (TP + TN) / (TP + TN + FP + FN) and is calculated per class where:

* True Positive (TP): defined as how many images were classified correctly (the actual value and the predicted value are the same).
* False Negative (FN): defined as the amount of times the model miss predicted the class (the sum of values of corresponding rows except the TP value).
* False Positive (FP): defined as the amount of times the model images as the class and was wrong (The sum of values of corresponding column except TP value).
* True Negative (TN) : defined as the number of times the model predicted the images as other classes and was right (the rest of the values).

### 3.2.2. Precision

It tells you what fraction of predictions as a positive class were actually positive. Precision = TP/(TP+FP)

### 3.2.3. Recall

It tells you what fraction of all positive samples were correctly predicted as positive by the classifier. It is also known as True Positive Rate (TPR), Sensitivity, Probability of Detection. Recall = TP/(TP+FN)

### 3.2.4. F1-score

It combines precision and recall into a single measure. Mathematically it’s the harmonic mean of precision and recall. F1-Score = 2\*Precision\*Recall/(Precision + Recall)

# 4. Literature review

## 4.1. Low-Light Image Enhancement Methods

Low light Image enhancement methods can be partitioned into three main categories. The first category methods are based on the histogram equalization (HE) (contrast adjustment using the images histogram). Dynamic histogram equalization (DHE) [1] - divides the histogram of the image into sub blocks and uses HE to stretch the contrast for each subblock. Adaptive histogram equalization (AHE) [2] - changes image contrast by calculating the histogram of multiple local areas of the image and redistributing the brightness.

The second category methods are Methods based on Retinex Theory [3] that assumes that an image is a combination of reflection and illumination. Such methods maintain the consistency of the reflectance, increase the brightness of the illumination, and take the pixel-wise product to enhance the low-light image. Low-light Image Enhancement via Illumination Map Estimation (LIME) [4] enhances a low-light image by estimating its illumination map. Single Scale Retinex (SSR) [5] aims to restore the brightness after Retinex decomposition. Multi-Scale Retinex (MSR) [6] combines the filtering results of multiple scales based on SSR, MSR adds a color recovery factor to tackle the color distortion caused by contrast enhancement in local areas of the image.

The third category is deep learning based methods– a great number of state-of-the-art methods have been developed for low-light image enhancement such as:LLNET [7] - a DP model for enhancing lightness and denoising images.

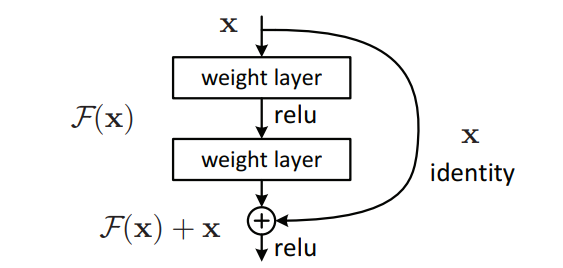
Branch Low-Light Enhancement Network (MBLLEN)][8] uses multiple subnets for enhancement and generates the output image through multi-branch fusion.

RetinexNet [9] - decomposes low-light input into reflectance and illumination and enhances the lightness over illumination.

## 4.2. Deep learning classification methods

Deep Residual Learning for Image Recognition (ResNet) [10] is a common neural network (NN) architecture used for deep learning computer vision applications like object detection, image segmentation and image classification. This network uses a technique called skip connections. The skip connection skips training from a few layers and connects directly to the output (figure 3).

The approach behind this network is instead of layers learning the underlying mapping, we allow the network to fit the residual mapping. This architecture allows a depth of up to 150+ layers that allow high accuracy and led to a breakthrough in the field of classification and identification accuracy. The method is still relevant and there are various improved methods of it such as ResNet200 [11]. Rethinking Model Scaling for Convolutional Neural Networks (EfficientNet) is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compoundcoefficient. Unlike conventional practice that arbitrarily scales these factors, the EfficientNet scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients. EfficientNets also transfer well and achieve state-of-the-art accuracy on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets, with an order of magnitude fewer parameters.



*Figure 3 - skip connection technique*

# 5. Methods

## 5.1. Block Diagram (Experiment and Implementation workflow)

1. Data augmentation

2. Low-light model benchmark

Evaluate accuracy on pristine images

Estimate accuracy loss on augmented images

Select dataset

Augment photo realistic low-light images

3. Image enhancement approach

4. Image augmentation approach

Evaluate accuracy improvement

Implement enhancement methods

Evaluate accuracy improvement

Implement and train

5. Comparative analysis and recommendation

## 5.2. Description of Project Phases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Title** | **Objectives** | **Subtasks** | **Expected outcome** |
| **1** | **Data augmentation** | Prepare dataset that contains synthetically generated low light images with known degradation parameters (e.g., SNR) | Select source image dataset | Low-light images dataset |
| Implement low light image augmentation model |
| Evaluate the quality of the augmentation |
| Generate augment dataset |
| **2** | **Low-light model benchmark** | Evaluate how state-of-the-art conventional models deal with low light images. Find the dependency between degradation parameters and loss of the accuracy | Evaluate accuracy on pristine images   * develop setup environment for deep learning * Train the model on the pristine images * Test the accuracy of the model for the pristine images | Baseline accuracy on pristine and low-light images of model |
| Estimate accuracy loss as function of degradation parameters |
| **3** | **Image enhancement approach** | Evaluate the improvement over benchmark that might be obtained by image enhancement applied prior to image classification as function of enhancement parameters and image degradation parameters | Research/Implement image enhancement methods   * Review the different solutions for low light image enhancement * Implement the chosen methods | Accuracy improvement as function of enhancement images |
| Evaluate accuracy improvement as function of enhancement and degradation parameters |
| **4** | **Image augmentation approach** | Evaluate performance of the model trained with low light synthetic images included in the training dataset | Design and implement training procedure | Accuracy improvement over the benchmark as function of degradation and model parameters |
| Evaluate accuracy improvement over the benchmark as function of degradation and model parameters |
| **5** | **Comparative analysis and recommendation** | Compare two approaches and provide a recommendation over applicability of each approach depending on the expected image degradation levels | Compare two approaches for various combinations of parameters | Recommendations over applicability of each approach depending on the expected image degradation levels |
| Summarize and document recommendations |

# 6. Engineering challenges

In this project there are a number of challenges.

First, we will need an environment for running experiments while changing and controlling the various parameters, this environment will need to run deep learning networks and support the following:

* Manage relatively large data sets
* Integrate different computer vision and image process libraries.
* Have the tools for analyzing the results of the tests.

In addition, the environment will need to support and track the different experiments:

* Organizing the different tests and document the results
* Manage and compare the tests results

# 7. Division of work between the partners:

|  |  |  |
| --- | --- | --- |
| **Task** | **Sub task** | **Assignee** |
| **Data augmentation** | Select source image dataset | Yarom |
| Implement low light image augmentation model | Yarom |
| Evaluate the quality of the augmentation | Yarom |
| Generate augment dataset | Yarom |
| **Low-light model benchmark** | Evaluate accuracy on pristine images | Rom |
| Estimate accuracy loss as function of degradation parameters | Rom |
| **Image enhancement approach** | Research/Implement image enhancement methods | Rom |
| Evaluate accuracy improvement as function of enhancement and degradation parameters | Rom |
| **Image augmentation approach** | Design and implement training procedure | Yarom |
| Evaluate accuracy improvement over the benchmark as function of degradation and model parameters | Yarom |
| **Comparative analysis and recommendation** | Compare two approaches for various combinations of parameters | Together |
| Summarize and document recommendations | Together |

# 

# 8. Required tools

## 8.1. Programs language

We will develop different parts of the projects using matlab and python. We will use the following python libraries:

|  |  |  |
| --- | --- | --- |
| **Tool** | **Description** | **Purpose** |
| Pandas | Open-source library providing high-performance, easy-to-use data structures and data analysis tools. | analysis tools |
| OpenCV | Open-source library that includes several hundreds of computer vision algorithms. | image processing tools |
| TensorFlow | Open-source platform for machine learning. | DP model tools |
| Matplotlib | Matplotlibis a comprehensive library for creating static, animated, and interactive visualizations in Python | analysis tool for visualization |
| Numpy | The fundamental package for scientific computing with Python | Numerical computing tools |

## 8.2. Develop environments and data analysis:

We will use the following work environments for development and analysis:

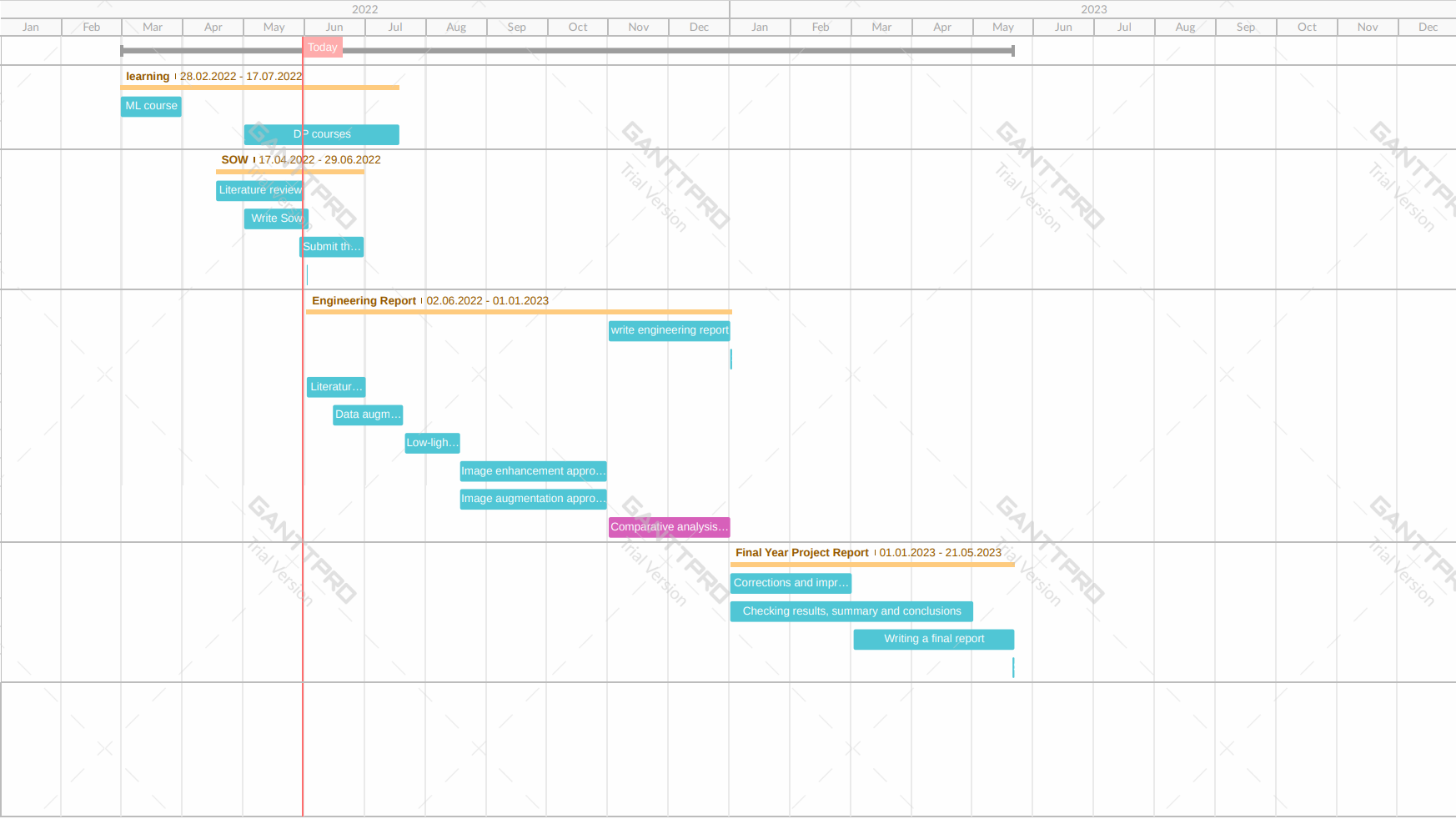
|  |  |  |
| --- | --- | --- |
| **Tool** | **Description** | **Purpose** |
| Microsoft Windows 10 | Operating system | work environment |
| GPU | Graphics processor unit | use for DP model calculations |
| python IDE | Code editor | used to write, edit and execute python code |
| Matlab | MATLAB is a programming and numeric computing platform | used to write, edit and execute matlab code |

# 9. Expected outcomes/deliverables

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Description** |
| D1 | Dark image augmentation algorithm | Python library for augmenting dark images from normal images |
| D2 | Augmented Dataset | A data set containing normal and augmented images for training and evaluation |
| D3 | Image enhancement lib | A python library containing different methods for enhancing low light images |
| D4 | Image classification lib | A python library implementation of several image classification pipelines |
| D5 | Image classification framework | A framework for low light image classification |
| D6 | Report and recommendations | An evaluation report and recommendation for selecting the best approach for low light image classification |

# 

# 10. Work plan



# 

# 11. References

1. Abdullah-Al-Wadud, Mohammad, et al. "A dynamic histogram equalization for image contrast enhancement." IEEE Transactions on Consumer Electronics 53.2 (2007): 593-600.‏
2. Pizer, Stephen M., et al. "Adaptive histogram equalization and its variations." *Computer vision, graphics, and image processing* 39.3 (1987): 355-368.‏
3. Land, Edwin H. "The retinex theory of color vision." Scientific american 237.6 (1977): 108-129.‏
4. Guo, Xiaojie, Yu Li, and Haibin Ling. "LIME: Low-light image enhancement via illumination map estimation." IEEE Transactions on image processing 26.2 (2016): 982-993.‏
5. Jobson, Daniel J., Zia-ur Rahman, and Glenn A. Woodell. "Properties and performance of a center/surround retinex." IEEE transactions on image processing 6.3 (1997): 451-462.‏
6. Jobson, Daniel J., Zia-ur Rahman, and Glenn A. Woodell. "A multiscale retinex for bridging the gap between color images and the human observation of scenes." IEEE Transactions on Image processing 6.7 (1997): 965-976.‏
7. Llnet: A deep autoencoder approach to natural low-light image enhancement, Pattern Recognition, vol. 61, pp. 650–662, 2017.
8. Lore, Kin Gwn, Adedotun Akintayo, and Soumik Sarkar. "LLNet: A deep autoencoder approach to natural low-light image enhancement." Pattern Recognition 61 (2017): 650-662.‏
9. Wei, Chen, et al. "Deep retinex decomposition for low-light enhancement." arXiv preprint arXiv:1808.04560 (2018).‏
10. He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.‏
11. Bello, Irwan, et al. "Revisiting resnets: Improved training and scaling strategies." Advances in Neural Information Processing Systems 34 (2021).‏
12. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.‏

# Data augmentation

## Datasets:

* + - 1. Lol dataset:
      2. Microsoft COCO is a large-scale object detection, segmentation, and captioning dataset published by Microsoft.

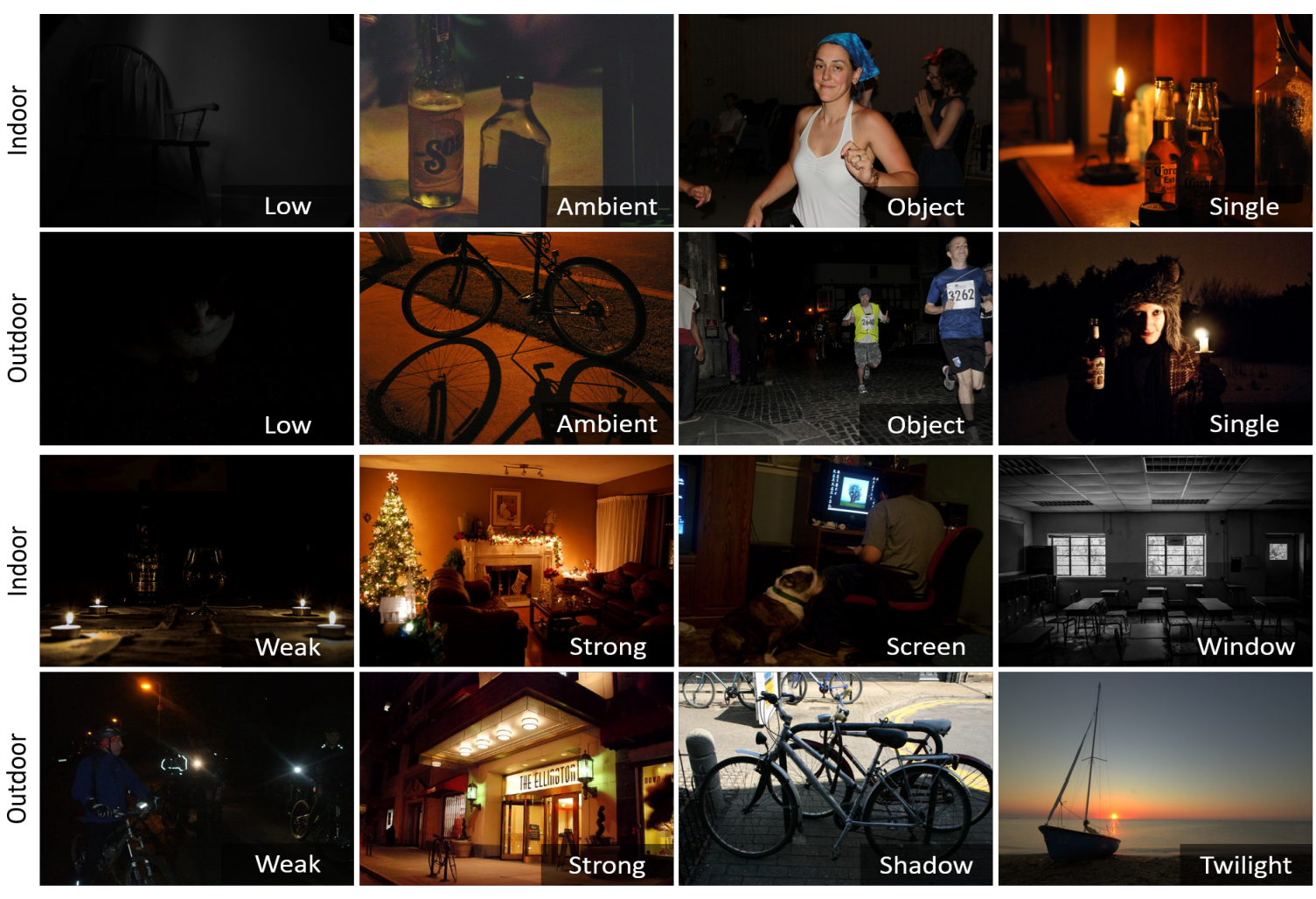
The dataset contains challenging, high-quality visual datasets for computer vision, mostly state-of-the-art neural networks And include the following features:

* Over 200’000 images of the total 330’000 images are labeled
* 1.5 Mio object instances
* 80 object categories, the “COCO classes”, which include “things” for which individual instances may be easily labeled (person, car, chair, etc.)

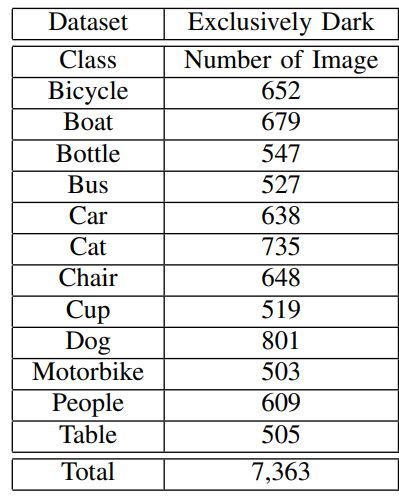
We extract relevant classes from coco dataset:

* + - 1. Exclusively Dark (ExDark) dataset (CVIU2019). The Exclusively Dark (ExDARK) dataset is a collection of 7,363 low-light images

It will used for train, validation and test



Types of class:



Types of low-light:

Chart, bar chart

Description automatically generated

• Low: Images with very low illumination and hardlyvisible details.

• Ambient: Images with weak illumination and the light source is not captured within.

• Object: Images where there is/are brightly illuminated object1(s) but surroundings are dark and the light source is not captured within.

• Single: Images where a single light source is visible.

• Weak: Images with multiple visible but weak lightsources.

• Strong: Images with multiple visible and relatively bright light sources.

• Screen: Indoor images with visible bright screens (i.e. computer monitors, televisions).

Window: Indoor images with bright windows as light sources.

• Shadow: Outdoor images captured in daylight but the

objects are shrouded in shadows.

• Twilight: Outdoor images captured in twilight (i.e. time of day between dawn and sunrise, or between dusk and sunset).

* + - 1. Microsoft COCO

## Implement low light image augmentation model:

In this section we implemented method for data augmentation for creating synthetic low light images.

In order to make low light images we used illumination change effects, we editing the entire image by altering the brightness, contrast, sharpness, saturation, noise.

### Gamma correction

Gamma correction can be used to correct the brightness of an image by using a non linear transformation between the input values and the mapped output values:

Diagram

Description automatically generated

As this relation is non linear, the effect will not be the same for all the pixels and will depend to their original value.

When γ>1, the pixels will be darker, and the histogram will be shifted to the left whereas it will be the opposite with γ<1.

### Brightness augmentation:

low light condition may instantly increase or decrease brightness. In order to model such illumination changes, we need to alter the pixels across the whole image

Brightness is adding constant value to all pixels in the image.

z – illumination noise

### Contrast augmentation:

The contrast of an image plays an important role in highlighting different objects in the scene. Low contrast images usually look softer and flatter, as well as lacking shadows and highlights. In reality, various occurrences can result in low contrast images. One of the common situations is lens flare in the image, where a bright light source scatters the light directly into the lens. Inspired by this observation, we propose a new data augmentation approach that varies the contrast of the image to improve the robustness of the framework. Specifically, we alter the contrast of the original image by applying the following formula:

C - contrast factor

**Sharpness augmentation (blur):**

Blurring can be achieved by convolving the original image with a n X n low pass use

## Color saturation

The color saturation of an image refers to the intensity of the color. The higher the saturation, the more colorful the image is. We edit the saturation of the whole image by converting it from the RGB to the HSV color space and directly changing the saturation attribute. Specifically, we scale the second

dimension of the HSV space which corresponds to saturation using a parameter s∈½0

s < 1, colour is diminished; conversely for s > 1 the colors become more saturated.

### Noise (gaussian)

TO DO: explain each parameter why it took

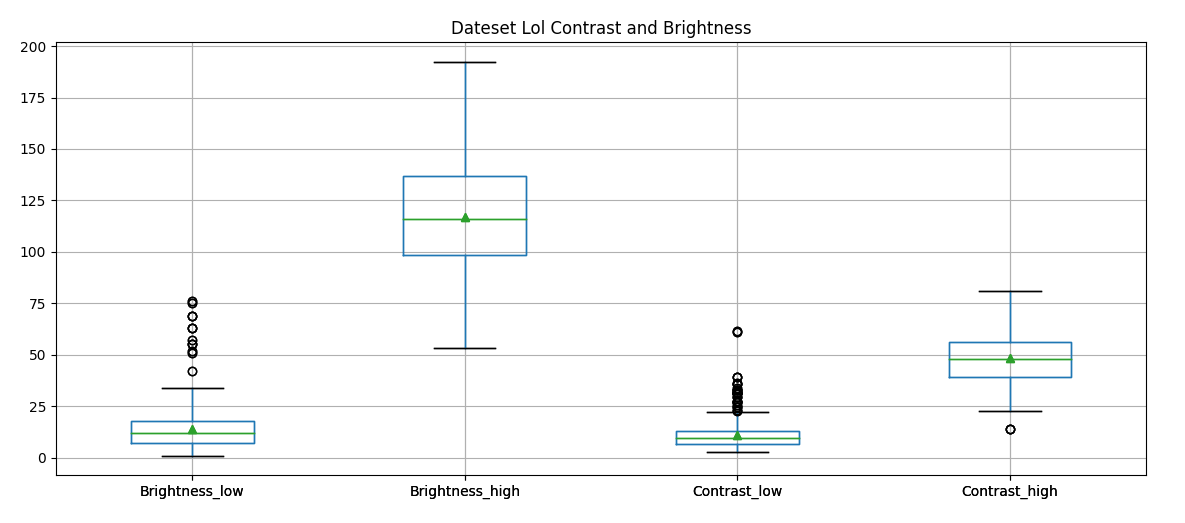
Method 2 :

noise formation model based on the characteristics of CMOS photosensors, thereby enabling us to synthesize realistic samples that better match the physics of image formation process.

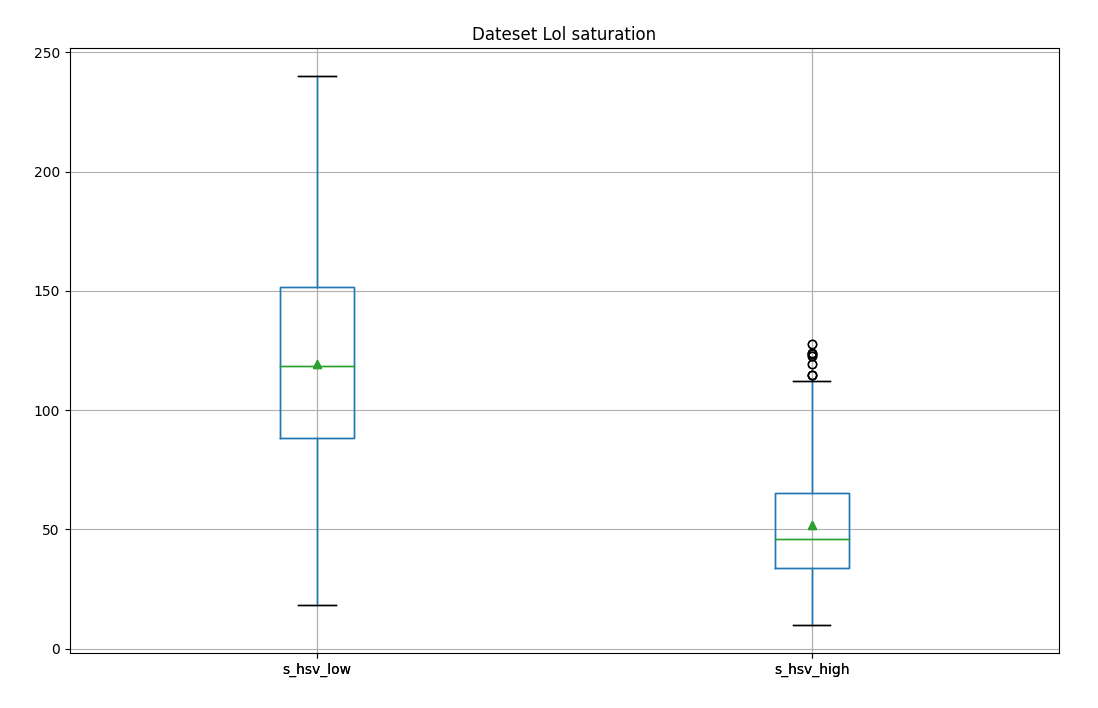
In order to define data augmentation real as possible we took a dataset (Lol) that contains low light and normal light images of the same object and extracted the contrast and sharpness and used it as input of our augmentation functions.

Statistic from the Lol dataset:

Brightness and Contrast differences between normal light and low light images



Saturation:



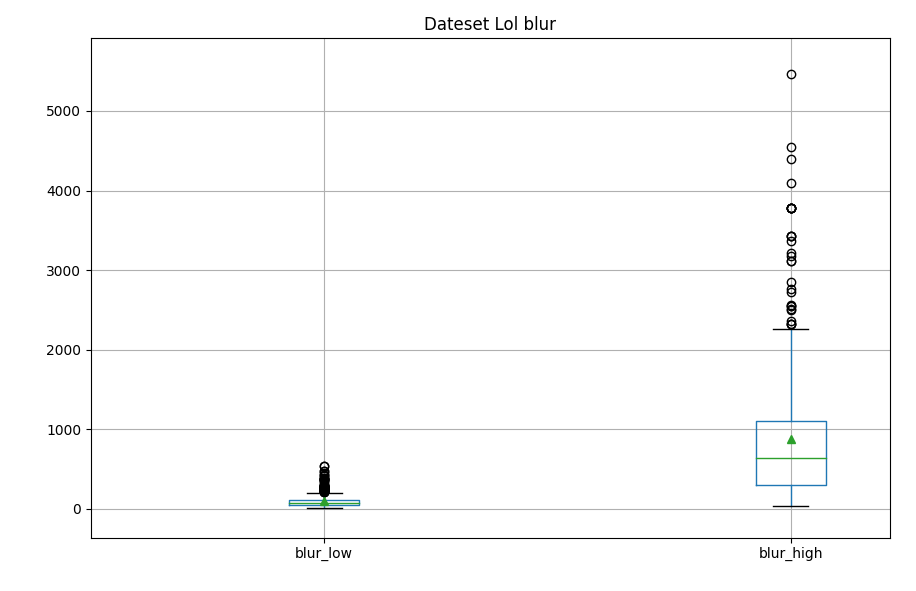
Estimate blur:

Detecting the amount of blur in an image

<https://pyimagesearch.com/2015/09/07/blur-detection-with-opencv/>

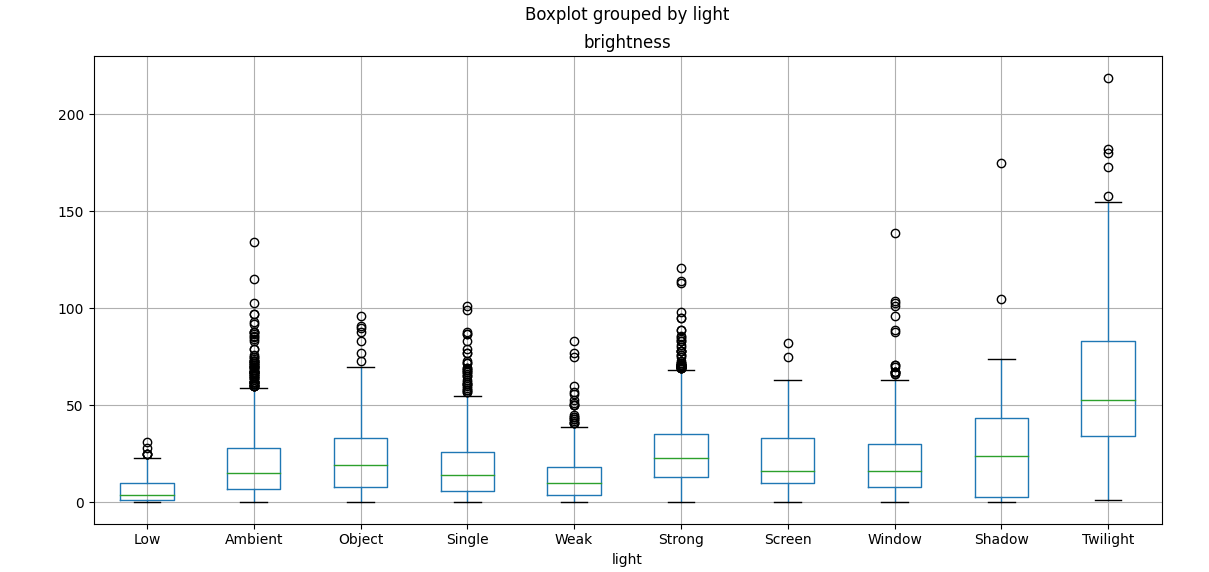
https://www.sciencedirect.com/science/article/abs/pii/S0031320312004736?via%3Dihub

Blur differences between normal light and low light images



1. Statistic from ExDark dataset for image augmentation:

Brightness:



Contrast:

Chart, box and whisker chart

Description automatically generated

Saturation:

Chart, box and whisker chart

Description automatically generated

Blur

Chart, box and whisker chart

Description automatically generated

# By analyze the Exdark and Lol datasets and for create different light condition we define input parameters as:

# Brightness changes – z ∈ (5, 80)

# Contrast changes – c ∈ (5, 80)

# Saturation changes – s ∈ (, )?

As apart of the data augmentation we random define the values of the parameters in the range we define.

1. **Low-light model benchmark**

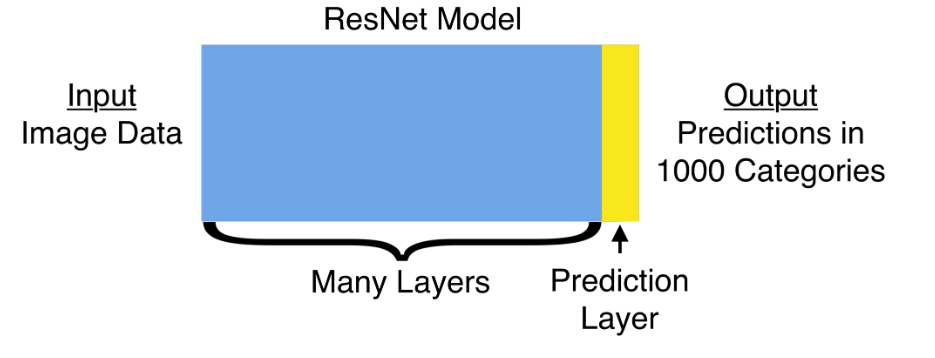
|  |
| --- |
| Evaluate accuracy on pristine images |
| Estimate accuracy loss as function of degradation parameters |

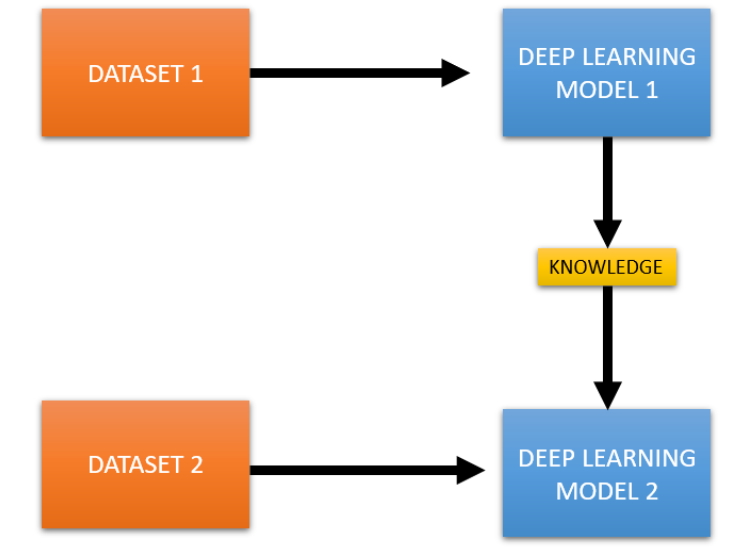
|  |
| --- |
| It is commonly agreed that CNN performs better when trained with more general data, i.e., very large numbers of images with complex variations. However, on account that the number of images in the ExDARK is still too small to train a full CNN model from scratch, we approach the task by fine-tuning and transfer learning.  The training setup of the experiments include replacing the last classification layer of the pre-trained Resnet-50 model which has 1,000 object classes for the ImageNet into the 12 object classes of the experimented dataset.  We trained the DP with the following parameters:  Learning rate: 0.00001 – it  Optimization scheme: Stochastic Gradient Descent  Batch size: 32  Prepressing: Cropping and filling  Epochs: 50 Classification Performance: Accuracy of Resnet-50 models trained using all relevant data from the datasets, with and without transfer train, different training ratios of bright images and lowlight images.  We did the following tests:   * + - 1. Pristine images: performance on COCO test images.       2. ExDARK: performance on ExDARK test images only       3. Syntetic Dark : performance on ExDARK test images only       4. Pristine and ExDark: performance on test images of both sets       5. Pristine and Dark       6. Overall: performance on test images of both sets  Evaluate accuracy on pristine images |
|  |

We cho

Transfer learning takes what a model learned while solving one problem (called a **pre-trained model**, because the model has already been trained on a different dataset), and applies it to a new application.

The very last layer makes predictions. We’ll drop in a replacement for this last layer of the ResNet model.





Classification Performance: