

**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

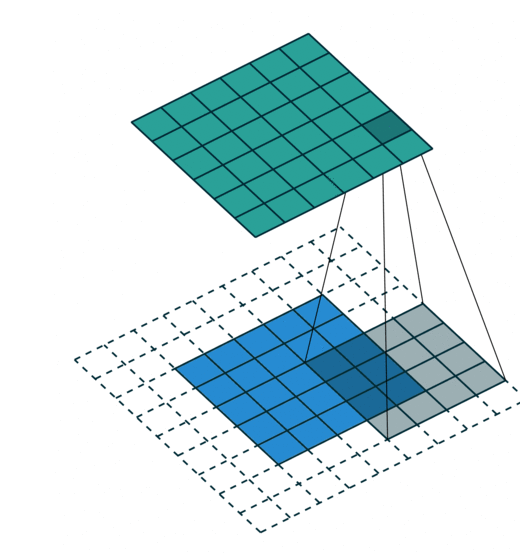
*Figure 1 – NN architecture example*

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. In addition to visual comparison 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

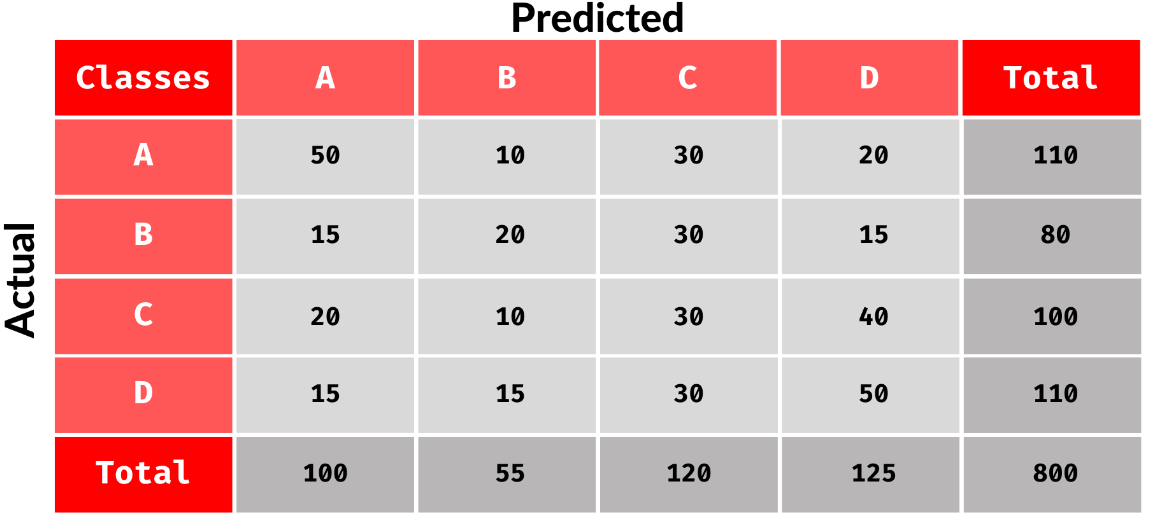
The different methods for classification will be measured by 4 measures:

* 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.

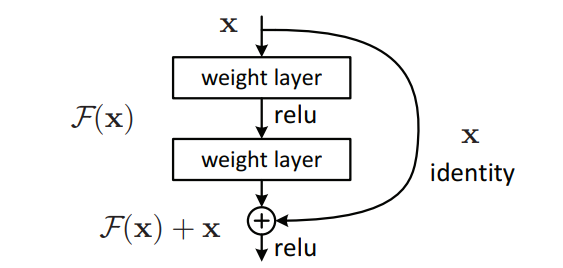
Branch Low-Light Enhancement Network (MBLLEN)][8] uses multiple subnets for enhancement and generates the output image through multi-branch fusion.

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 |

## 5.3. Project phases in depth

### Phase 1 – Data augmentation

#### Dataset

We chose ImageNet dataset, this dataset is a large database of images that is commonly used for training computer vision models. It contains more than 14 million images belonging to over 22,000 different classes, such as animals, objects, and scenes. ImageNet is often used in machine learning and deep learning research to develop and test algorithms for image classification and object detection tasks. In addition, we chose to use LOL (LOw-light) dataset which contains normal and low light images in order to evaluate the dark image augmentation algorithm.

#### Augment photo realistic low-light images

We propose a low light image simulation method to synthesize realistic low light images from normal light images. Low light images differ from normal images due to some dominant features: low brightness/contrast, noise, and sharpness. We tried to fit a transformation to covert the normal image to the low light image by analyzing images with different low light conditions. We find that the combination of linear parameters that change the contrast and gamma correction can approximate do this job well. The low light image simulation pipeline (without additional noise and blurring) can be formulated as:

Where is the synthetic low light image. is the normal light image. α is linear transformation that effect on contrast. is the gamma correction parameter used to correct\change the brightness of an image. is contrast parameter before gamma correction. is contrast after gamma correction for get image darker and blur with more difficult conditions.

As for noise we add Gaussian noise that simulate noise that happened due to the level of lighting. is variance

As for Blurring we use gaussian blur filter the filter does smooth images

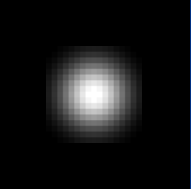


Figure Gaussian 2d window

Smoothing depend on – small low smoothing and large strong smoothing

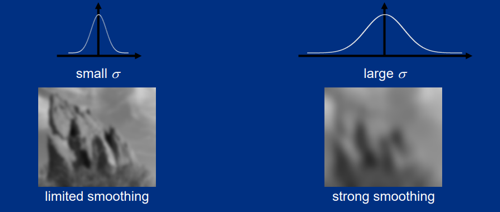


Figure gaussian filter: variance parameter effect

Blurring the image can be achieved by convolving the original image with gaussian blur filter:

Below our low light transformation algorithm pipeline:

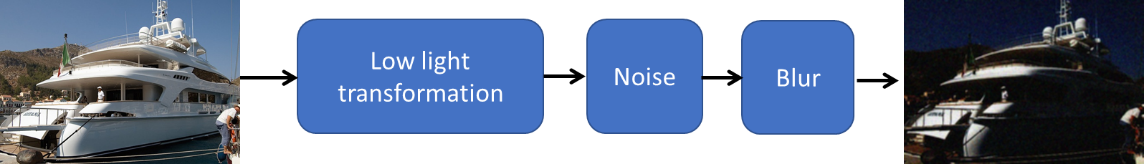


Figure - low light transformation algorithm pipeline

#### Evaluation and turning parameters

By analyze the results and compare to real low light images we get the following distribution for the parameters:

, 0.001, 0.01), ,

Below result with different parameter with normal image that has approximately the mean brightness and contrast of our dataset:

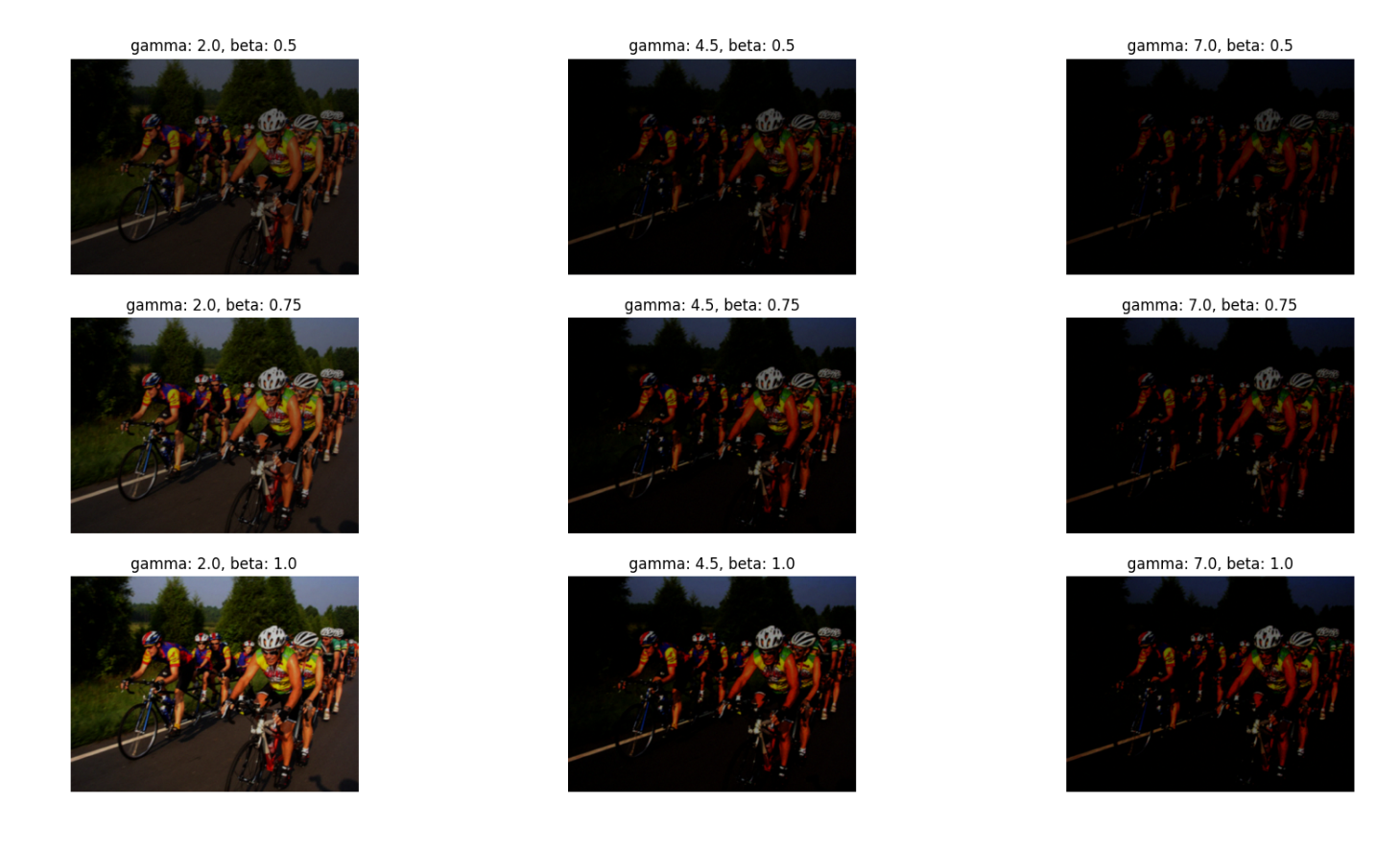


Figure - image augmentation with different parameters

#### Evaluation

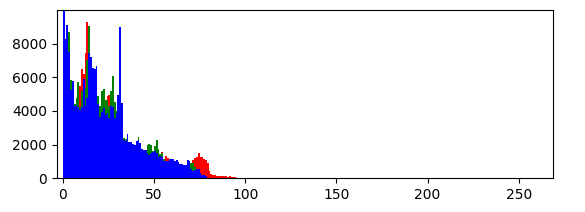
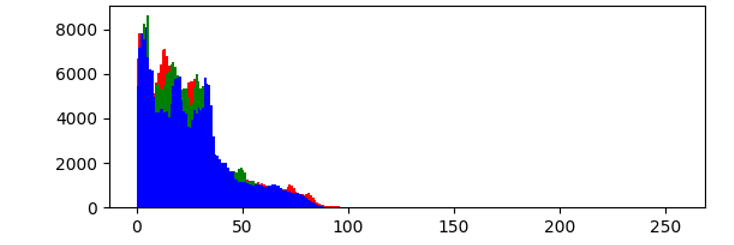
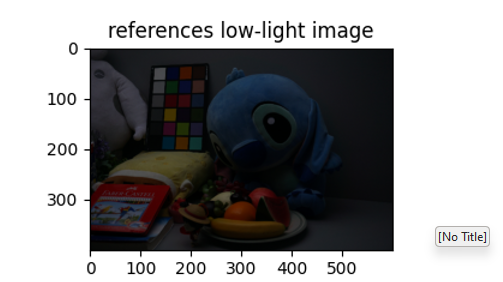
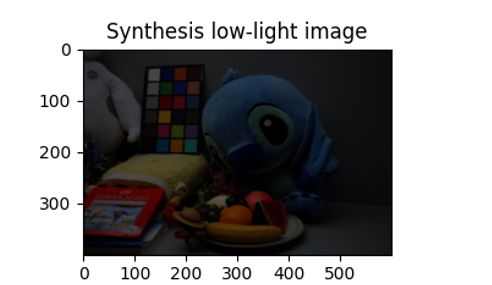
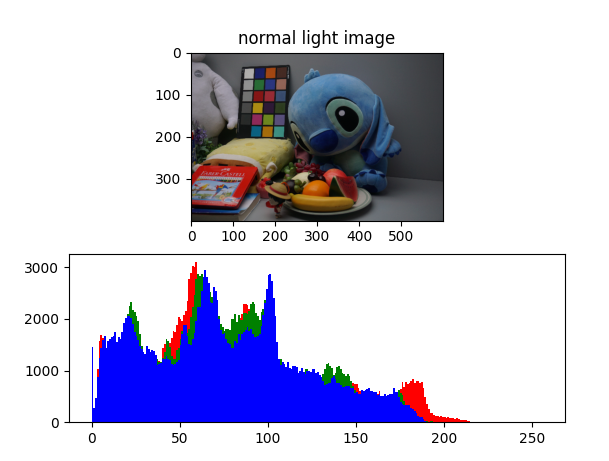
We verify the low-light transformation by comparing the algorithm output to real image taken in low-light conditions. we used visual comparison fig.5 and histogram comparison fig.6 of Y channel in YCbCr (Y is luminance\light intensity) between synthetic image and real dark image.

Chart, histogram

Description automatically generated

Figure - Y channel comparison of one example

Figure - one example of visual comparison



From the fig.5 and fig.6 we can conclude the augmentation yielded image that are approximately close to real low light images.

#### Low light augmentation levels

For our testing we created 4 levels of dark images (high level more difficult conditions) with the following parameters distribution:

level 1:

, 0.001, 0.01), ,

Level 2:

, 0.001, 0.01), ,

Level 3:

, 0.001, 0.01), ,

Level 4:

, 0.001, 0.01), ,

Below example from the dataset:

Figure - normal images



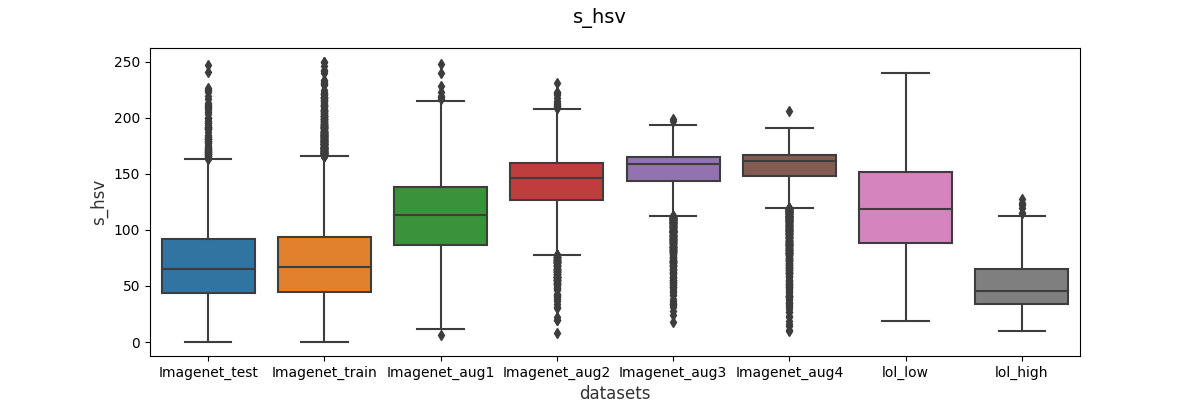
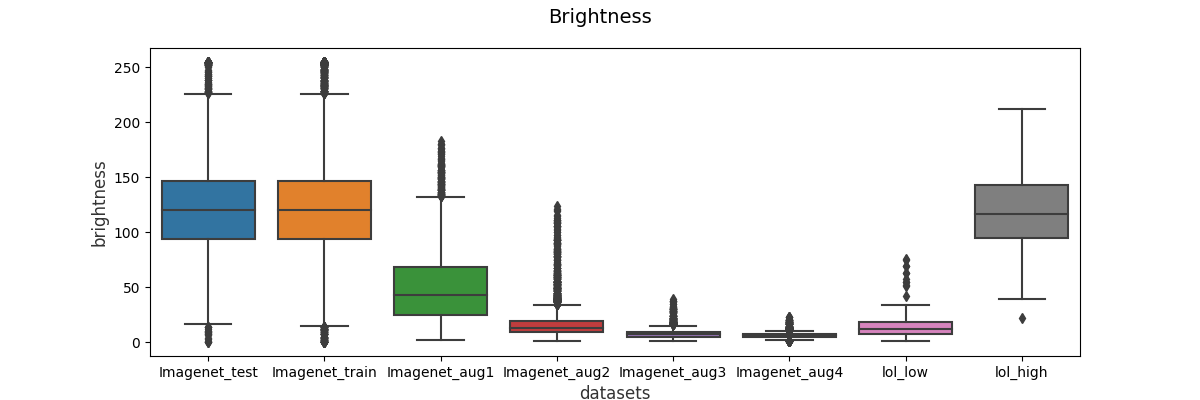


Figure - Low light augmentation levels exmaple

#### Evaluate datasets scope degradation parameters:

In this section we tested our generated datasets. we extracted for each dataset contrast, brightness, luminance, and saturation. And compare it to normal images datasets and to Lol dataset contains images of the same scene taken in normal and low light conditions.

Brightness and contrast – this parameter vastly decrease both in level and in range in dark scenery.

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

**Phase 2 – Low-light model benchmark**

#### Training approach

We approached the task by using fine-tuning because this technique is commonly used when the size of the new dataset is limited, As the pre-trained model provides a good starting point for the task and can help to improve the model performance. Fine-tuning typically involves adjusting the hyperparameters and fine-tuning some of the layers in the pre-trained model, rather than training the model from scratch. This can help to avoid overfitting on the new dataset and can often lead to better performance than training the model from scratch on the new data.

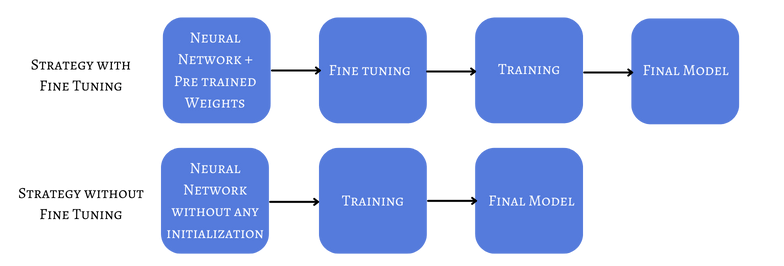


Figure - fine-turning strategy

#### Classification model

We chose ResNet pertained on ImageNet as our model because it is a convolutional neural network (CNN) architecture that has been shown to outperform many other popular CNN architectures on a wide range of benchmark datasets. ResNet is commonly used in image classification, and is notable for its use of skip connections, which allow the network to learn hierarchical representations of the input data. we decided to split our dataset as following 60% training, 20% validation, and 20% test and randomly select images and assign them it is standard split percentage. The model architecture we choose consists of the base ResNet50 pretrained model followed by a couple of dense layers for classification.

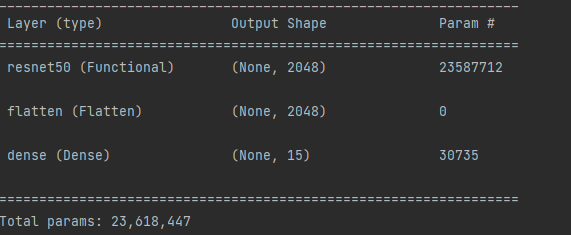


Figure - Model params

Table

Description automatically generated

Figure - Resnet architecture

#### Hyper parameters tuning

To improve performance on our dataset we adjusted the parameters in our ResNet model. These parameters included the learning rate, batch size, loss function, and optimizer. The **learning rate** is a hyperparameter that determines the step size at which the model updates its weights during training. Turning the learning rate up or down can affect the speed at which the model learns, as well as its final accuracy. A higher learning rate can lead to faster learning but can also cause the model to converge at a suboptimal solution or even diverge. A lower learning rate can result in slower learning but may lead to a better model. We chose to use learning rate scheduling, start with a relatively high learning rate, and then decrease it over time if the model is not making sufficient progress. The **batch size** is a hyperparameter that determines the number of samples that are processed by the model in each iteration of training. Choosing the right batch size for a ResNet model can have a significant impact on the model's performance. A larger batch size can lead to faster training and better utilization of the available hardware but can also make the model more sensitive to noise in the data and can make it more difficult to find a good learning rate. A smaller batch size can lead to slower training but can also provide a more stable learning signal and make it easier to find a good learning rate. In general, we start with a relatively small batch size and then increase it if the model is not making sufficient progress. In our model we get the best performance with batch mark of 32. As for loss function, we used a common loss function for image classification task called Cross-entropy. Cross-entropy loss measures the performance of a classification model whose output is a probability value between 0 and 1 for each class. Cross-entropy loss increases as the predicted probability diverges from the actual label. A perfect model would have a log loss of 0. As for **the optimizer** we chose to use Adam (Adaptive Moment Estimation) is our optimization algorithm. It is an adaptive algorithm that uses an exponentially decaying average of past gradients to scale the learning rate. This allows it to adapt to the changing nature of the gradients as the model is trained and can help to improve convergence and avoid getting stuck in local minima. This allows Adam to be more computationally efficient and effective for training large and complex models.

#### Standard augmentation

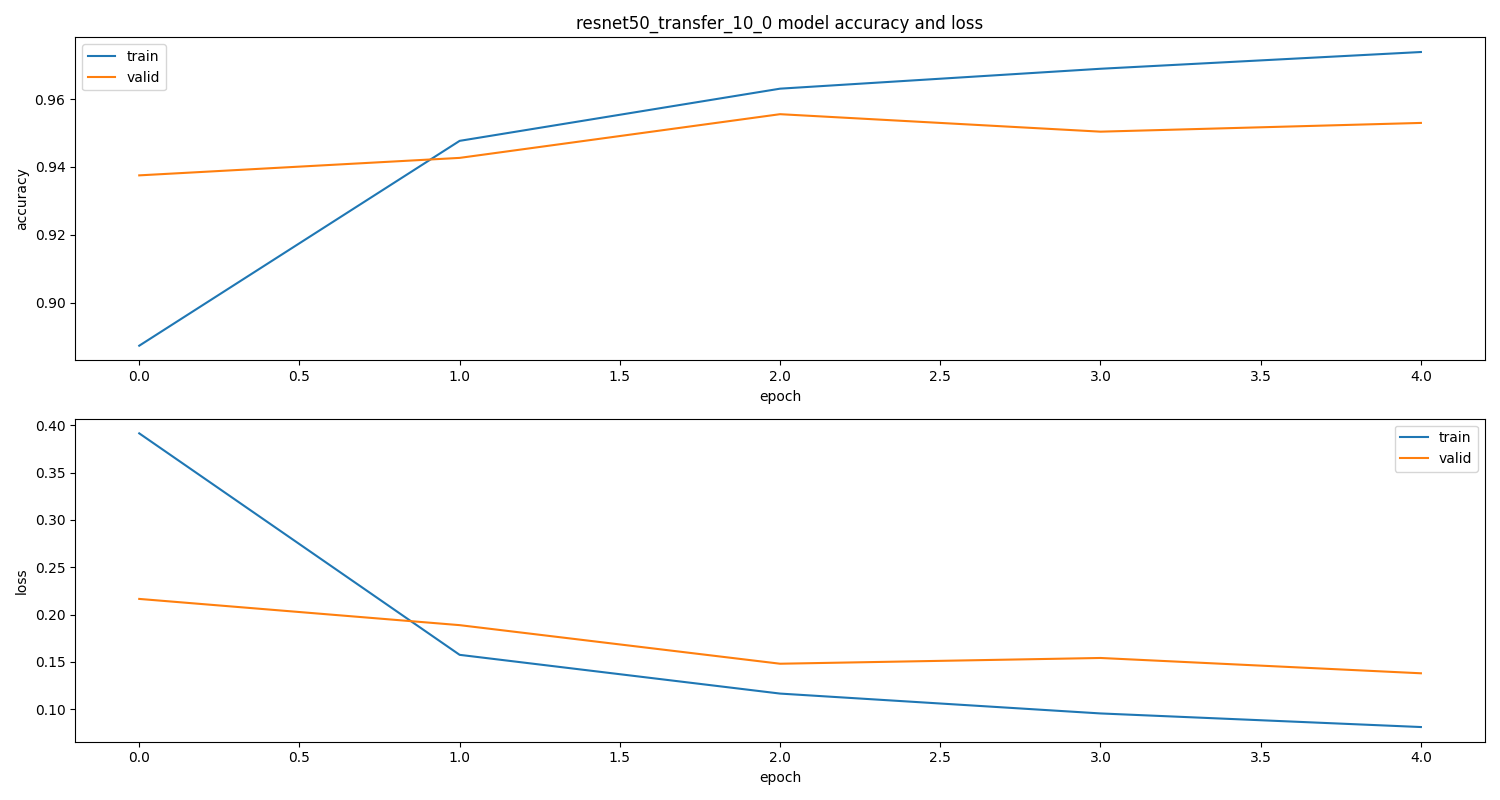
Data augmentation is a technique used to generating modified versions of the original data. This can help improve the performance of a machine learning model by providing it with additional training examples to learn from, which can help prevent overfitting and improve generalization to new data. We chose to use random horizontal and vertical flips augmentation, this can help the model learn to recognize the same object in different orientations, which can be important for achieving good performance on real-world data where the orientation of objects may vary.

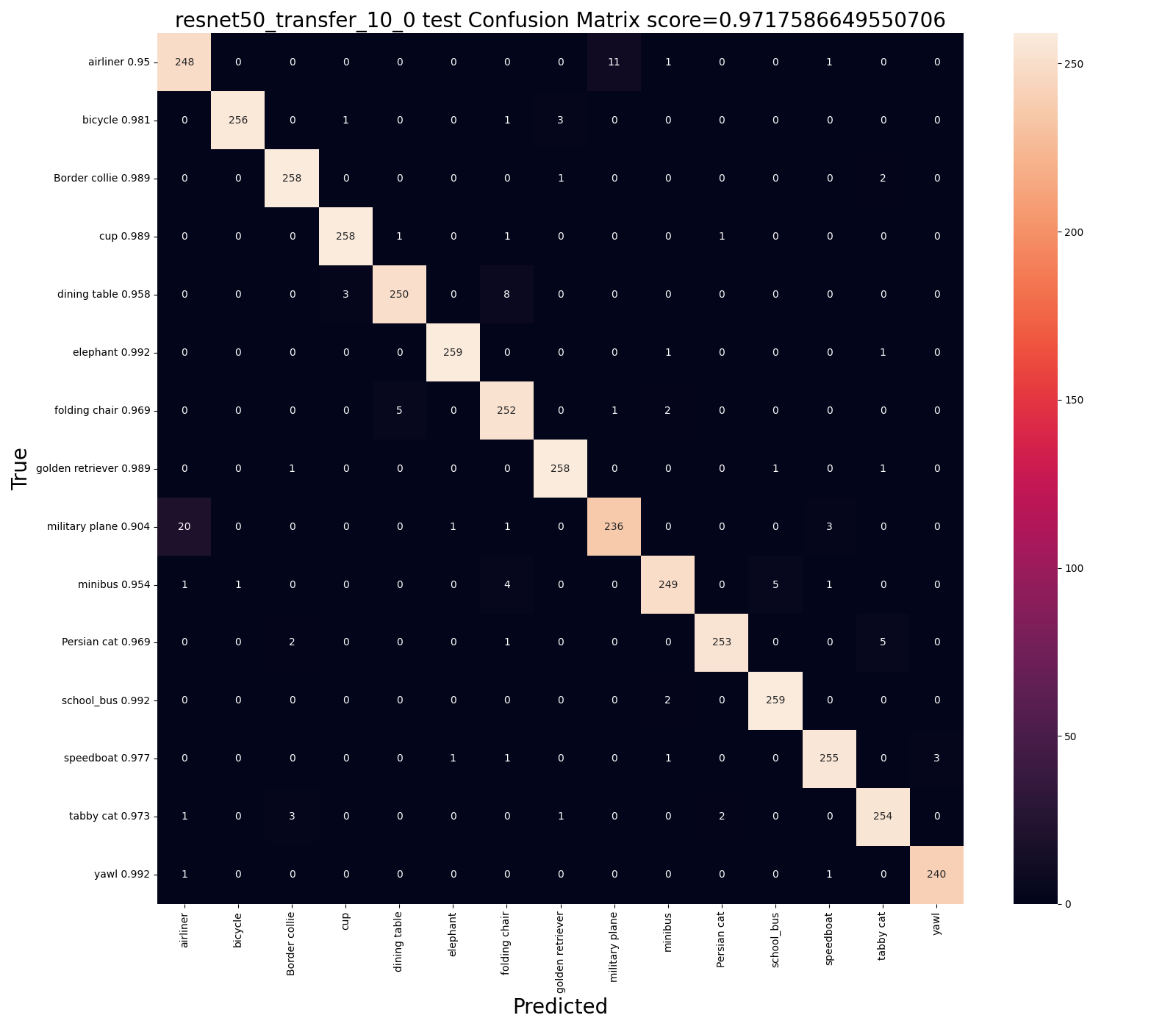
#### Evaluate accuracy for pristine images

We chose a total of 15 classes from the ImageNet dataset and trained our model on them using the hyper parameters mentioned above tableX, We ran 50 epochs and take the best one with the lowest loss. The model accuracy results:

Validation dataset: accuracy of 0.95 and loss of 0.15

Test dataset contain 5 datasets:We created confusion matrix of the results

Test set: accuracy of 0.97, each class accuracy can be seen FigX



#### 

#### Evaluate loss of accuracy for low light images

We tested our trained model (trained on pristine images) on augmented low light images with different levels as we defined at LINK.

Results:

Level 1: Accuracy X FIGX

Level 2: Accuracy X

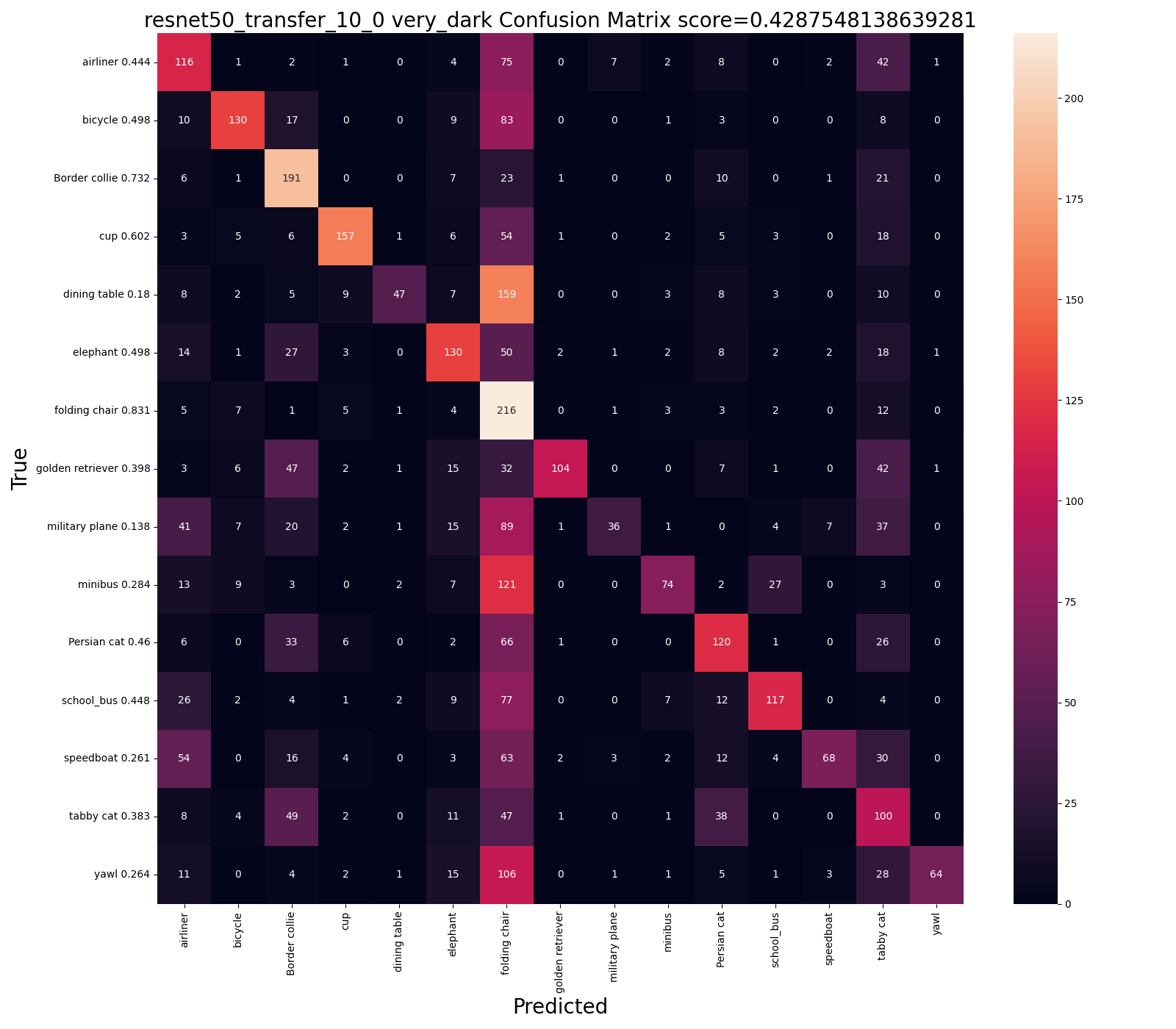


FIG FIG FIG

SUMMERY

# 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:

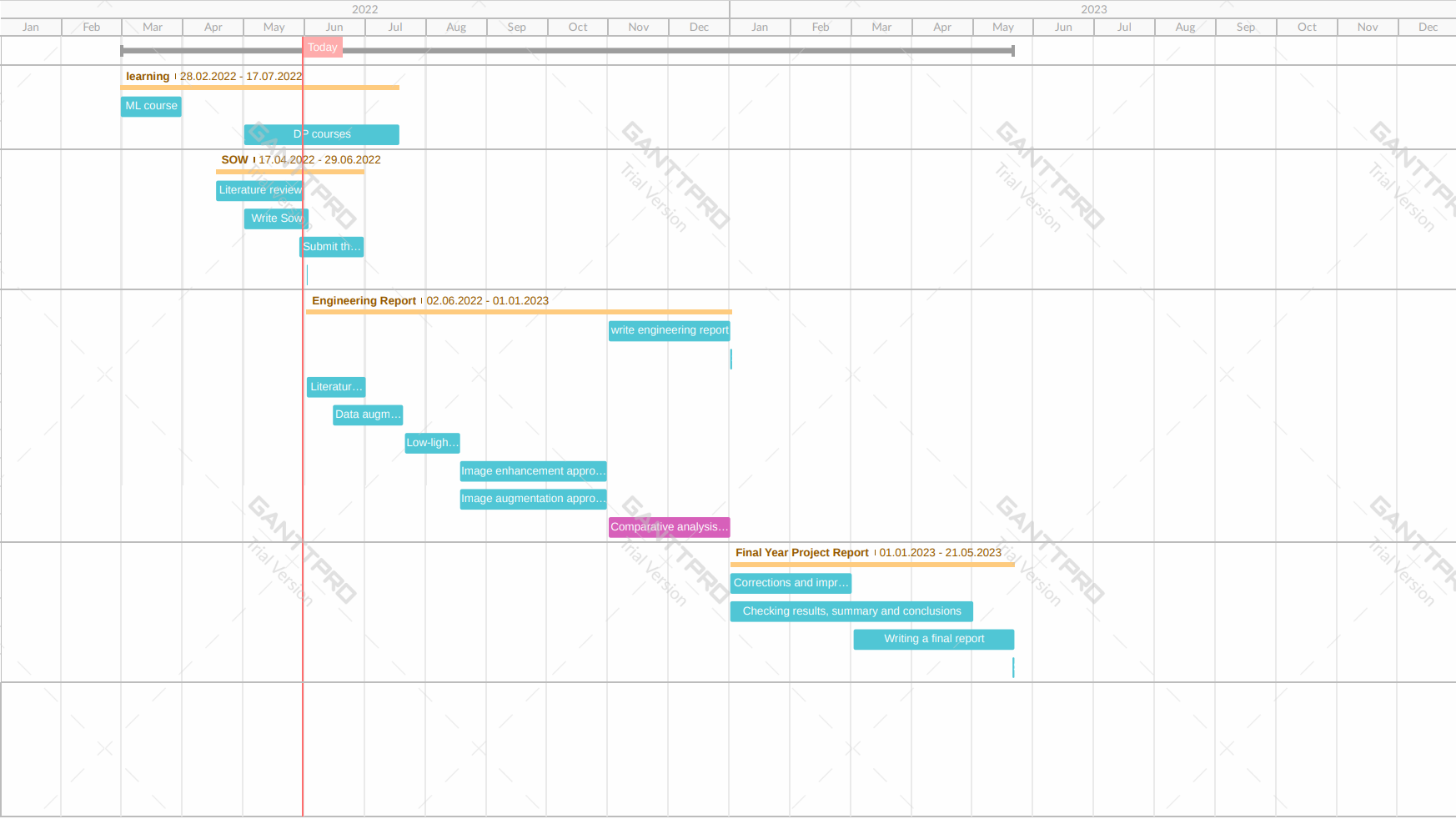
|  |  |  |
| --- | --- | --- |
| **Tool** | **Description** | **Purpose** |
| Microsoft Windows 10 | Operating system | work environment |
| GPU | Graphics processor unit | use for DP model calculations |
| python IDE | Code editor | used to write, edit and execute python code |
| Matlab | MATLAB is 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. 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 200000 images of the total 330000 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 chose to use some of the classes from the dataset as follows:

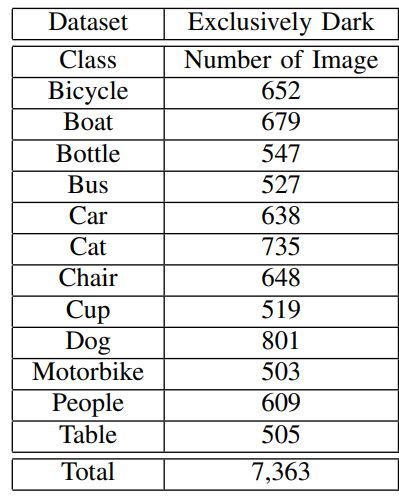
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **labels** | bicycle | boat | bottle | bus | car | cat | chair | cup | dog | motorcycle | dining table | person |
| **image count** | 5151 | 5104 | 5084 | 5096 | 4948 | 4634 | 5051 | 4916 | 5047 | 5047 | 5078 | 5093 |

This data set will be used to train the model as well as create augmented images.

* + - 1. Exclusively Dark (ExDark) dataset (CVIU2019). The ExDark dataset is a collection of 7,363 low-light images. This dataset contains a variety of real low light images of the following classes:

Types of class:

A picture containing text, different, same

Description automatically generated

The data set contains different types of low light images divided by categories as follows:

• Low: Images with very low illumination and hardly visible details.

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

• Object: Images where there is/are brightly illuminated object 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 light sources.

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

• Screen: Indoor images with visible bright screens

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).

Chart, bar chart

Description automatically generated

Figure low light images types distribution

This data set will be used to test and train in order to include real dark images.

## Implement low light image augmentation model:

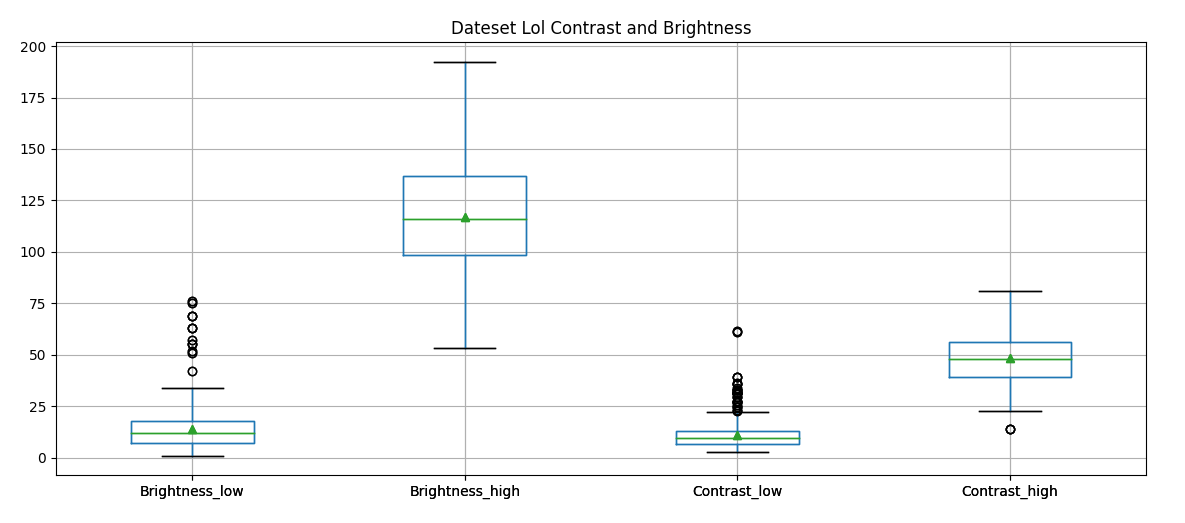
In this section we implemented method for data augmentation for creating synthetic low light images.

**Get to know low light images**:

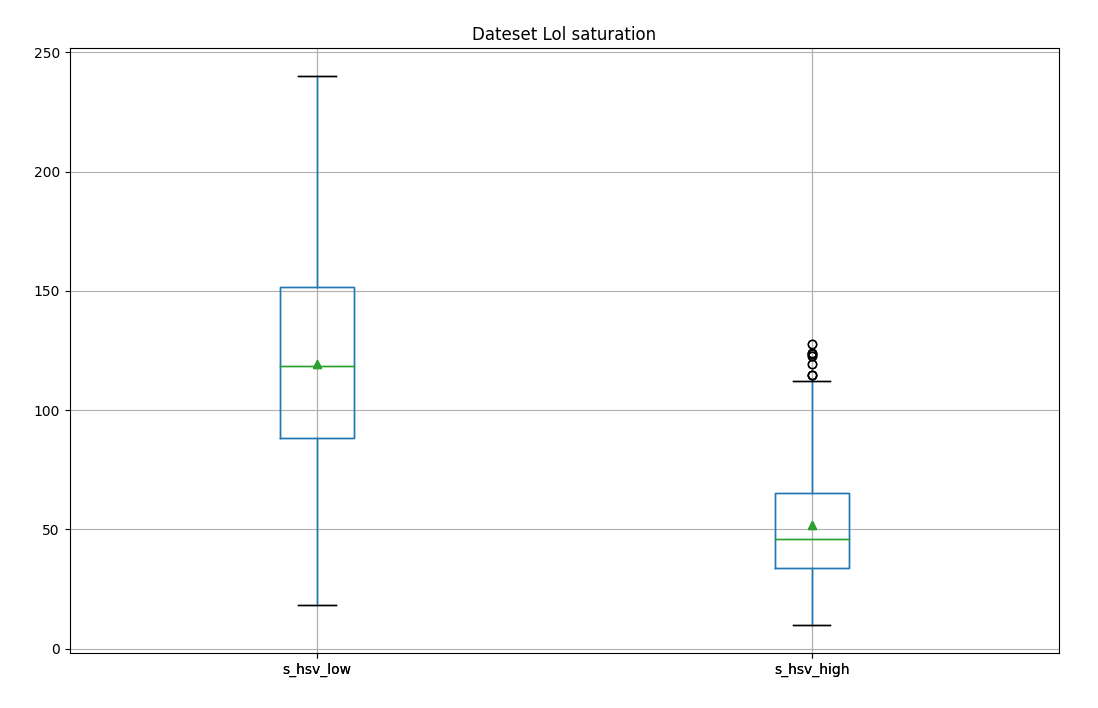
Analyze low light images

Lol dataset contains images of the same scene taken in normal and low light conditions.

In order to define data augmentation parameters as real as possible we took the Lol dataset and extracted the contrast, brightness, saturation and sharpness to compare between the image pairs.



Brightness and contrast – this parameter vastly decreases both in level and in range in dark scenery.

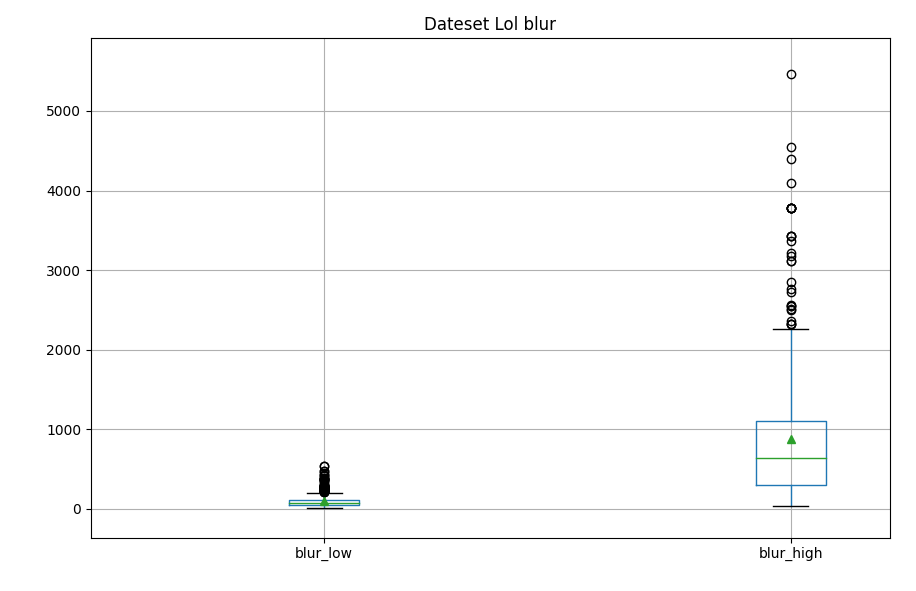


Estimate blur:

Detecting the amount of blur in an image

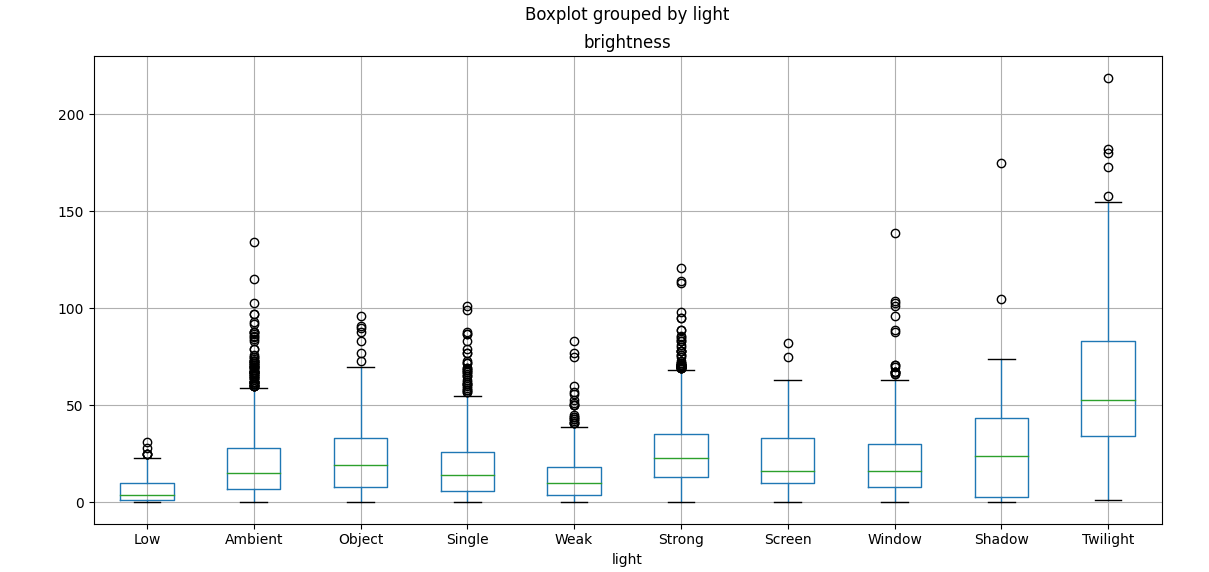
<https://pyimagesearch.com/2015/09/07/blur-detection-with-opencv/>

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



Blur – low light images are less sharp, this is the result of the camera shutter being open for longer time in order to absorb more light.

1. ExDark dataset for image augmentation in this section we evaluate some type of low light affects:



Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated