Weekly Report

Comparative Analysis of Image Classification Models for Efficient and Accurate Classification across Diverse Image Types

Student: Yusif Mukhtarov

Date: 07/03/2023

Measurement

Considering the fact that the current research project is related to classification models, it falls under the **nominal** measurement type. Moreover, accuracy, recall and precision will be used to assess different classification models on data with different noise levels and degree of rotation. Nominal measurements are characterized by values that identify attributes uniquely, where the values are non-ordered. In the case of classification metrics like accuracy, precision, and recall, the values represent different categories or classes, such as "true positive," "true negative," "false positive," and "false negative." These categories are discrete and do not have an inherent order or magnitude. Accuracy, which is ratio of correct classification to the total number of samples will be used as main metric. However, recall and precision will be also used for several reasons:

1. Class Imbalance: When the distribution of classes is imbalanced, accuracy alone can be misleading. Recall focuses on capturing the instances of the minority class, making it useful in imbalanced scenarios.
2. Asymmetric Costs: Misclassifying instances of one class may have more severe consequences than misclassifying instances of another class. Precision measures the model's ability to avoid false positive predictions, which is important when the cost of false positives is high.
3. Trade-off between Recall and Precision: Increasing recall often decreases precision and vice versa. Evaluating both metrics allows for a comprehensive understanding of the model's performance and the trade-off between capturing all positive instances and avoiding false positives.

To calculate recall and precision for image classification with 15 classes, the following steps will be made:

True Positives (TP): The number of images that were correctly classified as belonging to each class should be counted. False Positives (FP): The number of images that were incorrectly classified as belonging to each class should be counted. False Negatives (FN): The number of images that were incorrectly classified as not belonging to each class should be counted. True Negatives (TN): The number of instances that were correctly classified as not belonging to the specific class should be counted.

Once TP, FP, FN, and TN values have been calculated for each class, recall and precision finally can be calculated as follows:

Recall measures the proportion of actual positive instances that are correctly identified. It is calculated as the ratio of true positives to the sum of true positives and false negatives for a specific class.

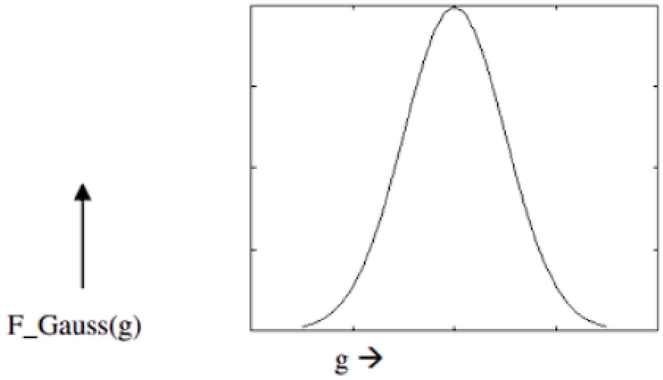
Recall = TP / (TP + FN)

Precision measures the proportion of instances that are correctly classified as positive out of all instances that are classified as positive. It is calculated as the ratio of true positives to the sum of true positives and false positives for a specific class.

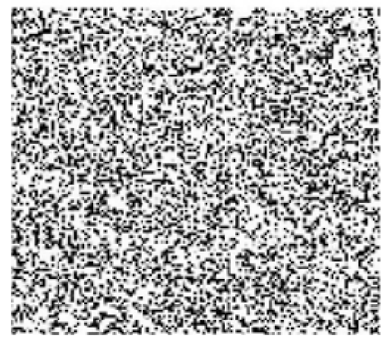
Precision = TP / (TP + FP)

Statistical Analysis and Visualization

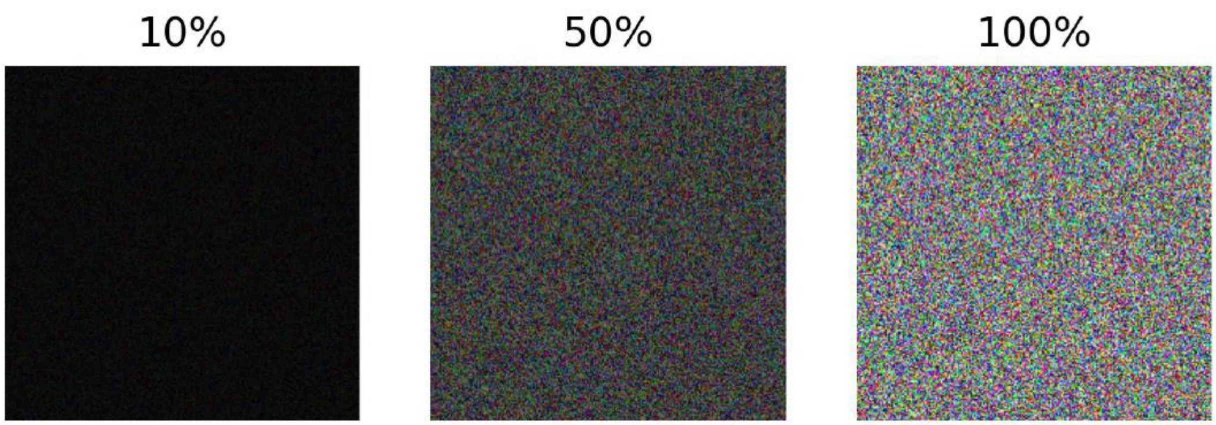
It is commonly known that Gaussian noise is statistical noise with a probability density function equal to the normal distribution. Gaussian noise has a uniform distribution throughout the signal. A noisy image has pixels that are made up of the sum of their original pixel values plus a random Gaussian noise value. The probability distribution function for a Gaussian distribution has a bell shape. below figure shows the Gaussian distribution function of Gaussian noise and pixel representation of Gaussian noise:

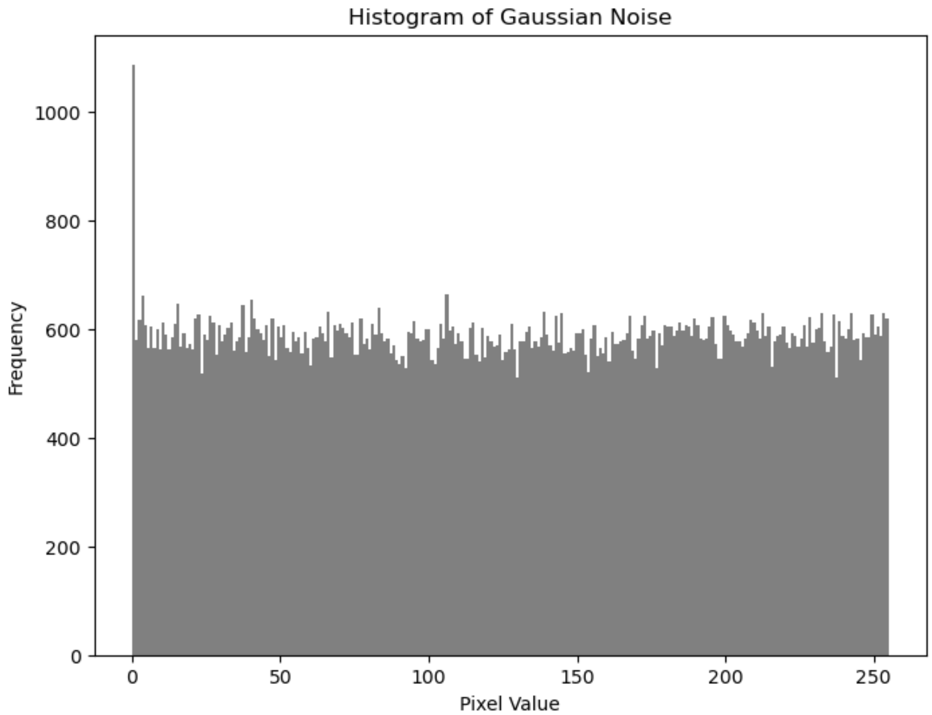


Gaussian Distribution function


Gaussian noise

In our project we have different levels of noise which look like:





All the visualization of the images with different noise levels have been shown in Report 2, and this part it is noteworthy to show DMAIC cycle of my project:

1. **Define**. In the picture below you can see the difference between the accuracies of model tested on images without noise and the same images with the noise:

```
Accuracy of the network: 15.533333333333333 %
Accuracy of Bean: 0.0 %
Accuracy of Bitter_Gourd: 95.0 %
Accuracy of Bottle_Gourd: 9.5 %
Accuracy of Brinjal: 0.0 %
Accuracy of Broccoli: 27.5 %
Accuracy of Cabbage: 0.0 %
Accuracy of Capsicum: 0.0 %
Accuracy of Carrot: 0.5 %
Accuracy of Cauliflower: 0.5 %
Accuracy of Cucumber: 0.0 %
```

```
Accuracy of the network: 99.66666666666667 %
Accuracy of Bean: 100.0 %
Accuracy of Bitter_Gourd: 99.0 %
Accuracy of Bottle_Gourd: 100.0 %
Accuracy of Brinjal: 99.5 %
Accuracy of Broccoli: 99.0 %
Accuracy of Cabbage: 99.5 %
Accuracy of Capsicum: 99.5 %
Accuracy of Carrot: 100.0 %
Accuracy of Cauliflower: 100.0 %
Accuracy of Cucumber: 99.0 %
```

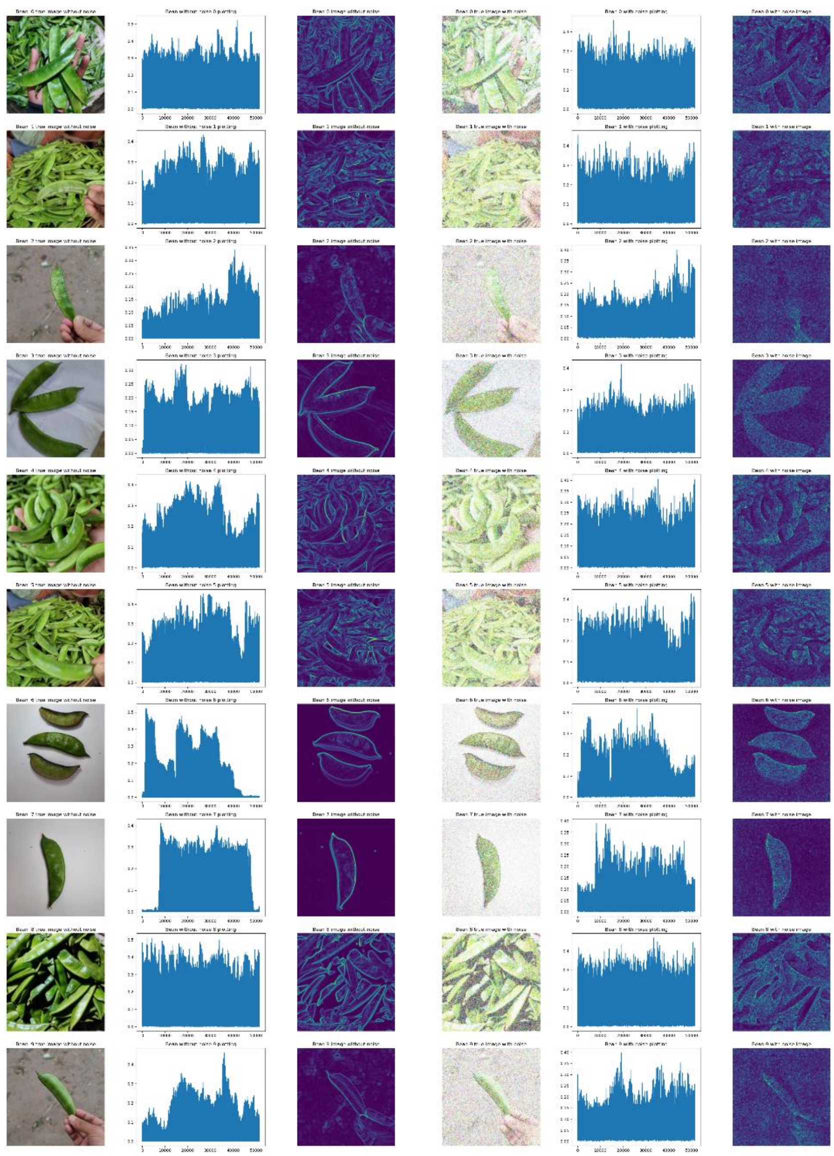
The main goal of experiment is to find a way to achieve the highest accuracy in classification of noisy images with different models, and then comparison of those models.

2. **Measure.** It was already discussed in the first part of this report.
3. **Analyze.** Careful analysis of the future extractions of models led us to specific conclusions. In the last CNN layer of all models (AlexNet, VGG16 and inceptionv3)

statistical similarities have been observed in the case of images with noise. However, in the case of noisy images data from the last CNN layers looks more random. Unfortunately, due to holidays SSH connection to the working environment had been lost, and most of those statistics are currently unavailable. As soon as the connection is restored, all those statistics will be added to this report in visual format.

4. **Solution and Improve.** To diminish the gap between those accuracies hyper tuning of model parameters will be done.
5. **Control.** The main way to control and track the progress is to use X-Bar, R-charts and other types of plotting to see the progress in accuracy, recall and precision scores. Moreover, validation and training curves will be also used to see the impact of different parameters during hyper tuning.

I have also tested several feature extraction methods and find out Sobel which showed the best results among all of others:



As is obviously seen from the picture (one in good quality is available in my github, statistical_analyses.ipynb) in most cases it is able to find out the patterns of vegetables. In histograms the pixel values after feature extraction are plotted (50000 on X axes means 227x227 pixels = 51529).