

# Model Description & Evaluation

For Phase 1 (Beeldherkenning9c\_v0)

Author: Kwan Suppaiboonsuk

Date: Monday, July 8, 2019

Last review date: Wednesday, August 7, 2019

## Contents

- Introduction ..... 1
- Dataset..... 1
  - Original..... 1
  - Augmented..... 2
- Training..... 3
- Evaluation ..... 4
  - Validation Dataset ..... 4
  - Performance - accuracy ..... 4
  - Performance – time..... 6
- Discussion ..... 7
  - Recommendations..... 7

## Introduction

For the Beeldherkenning project, an image classification model has been trained for 9 different classes of objects. This document aims to explain the process that was completed in order to get the model, as well as an evaluation of the model's performance.

There are three main parts to describe the model: the dataset, the training, and the evaluation. These will be covered in the section below.

At the end of the document, the results of the performance is discussed. Recommendations are given for how a future model could be trained to achieve a better performance – higher classification accuracy within a shorter amount of inference time.

## Dataset

A model is as good as it's dataset. This section will describe the dataset that was generated for retraining the object classification model, as well as how the dataset was made.

Altogether there are three different datasets: the original dataset, the augmented dataset, and the validation dataset.

The original dataset is a smaller dataset of images that were manually captured.

The augmented dataset uses the original dataset to generate more images. This dataset is used as input for retraining the pre-trained model, with a 70/30 training-testing set split.

The validation dataset is also based on the original dataset. A different set of randomly augmented images are generated and used to evaluate the performance of the model. The process for this dataset will be explained under the Evaluation section.

Table 1 gives an overview of the total amount of samples in each of the three datasets.

**Table 1. Overview of datasets**

Dataset	Total Amount of Image Samples
<b>Original</b>	705
<b>Augmented</b>	9000
<b>Validation</b>	7755

## Original










Images in the original datasets were collected manually by photographing the objects from the top view against various backgrounds. The camera used was a phone camera (Samsung Galaxy S9). Apart from direct top-view capture, photos were also captured from slight angle changes. As well as under various lighting environments.

The images are then rescaled to 299x299 px. There is a total of 705 images.

In Figure 1, an overview of the image dataset can be found. The table on the left-hand side gives the translation between the class label and the product info (Modderkolk article number and GTIN product

code). The grid of images on the right shows a few of the samples from each class label, with each row showing a different label. The rows are in order from 1 to 9.

**Figure 1. Overview of original dataset**

	Description - MK	Description -GTIN	Sample Amount	
1	111325	4046356482561	66	
2	71890	4017918929145	69	
3	101173	4017918188030	77	
4	32739	3250615721860	66	
5	69440	3593150039763	76	
6	92274	3250615510754	107	
7	71309	4015081184880	82	
8	97856	4011209783942	69	
9	116813	4011209781542	93	



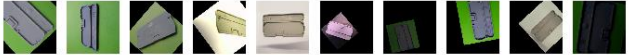


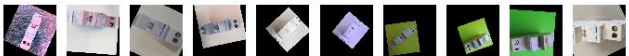
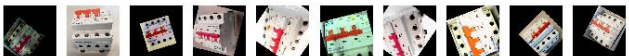

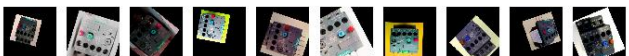
### Augmented

The augmented dataset has a total of 9000 images. Within this amount, the dataset also contains the whole original dataset. The rest of the images are randomly generated using the py-image-dataset-generator tool. The combination and amount of augmentation is randomized between rotation, blurring, Gaussian noise, hue, and contrast. Some images are also randomly scaled.

This dataset is used as input for retraining the Tensorflow model. Within the training, there is also a randomized training and testing set split. A split ratio of training and testing set performed on this dataset is 70:30.

In Figure 2, the overview of the augmented dataset can be seen.

Figure 2. Overview of augmented dataset

	Description - MK	Description -GTIN	Sample Amount	
				
				
1	111325	4046356482561	1000	
2	71890	4017918929145	1000	
3	101173	4017918188030	1000	
4	32739	3250615721860	1000	
5	69440	3593150039763	1000	
6	92274	3250615510754	1000	
7	71309	4015081184880	1000	
8	97856	4011209783942	1000	
9	116813	4011209781542	1000	
				

## Training

The method of training used is transfer learning. This utilizes a model that has been pre-trained on a larger dataset and re-trains the last layers of the model to recognize the 9 objects.

The pretrained model used is from Tensorflow – the SSD MobileNet V1 model, trained on the COCO dataset. [ssd\_mobilenet\_v1\_coco\_2017\_11\_17]

The benchmark values for this model is:

Speed	COCO mAP <sup>[^1]</sup>
30ms	21

The COCO dataset contains 330K images with object categories and 91 stuff categories of general everyday things (eg. Person, dog, cat, plane, etc.)

The retraining uses the augmented dataset with a split of 70/30 between the training and the testing set. The batch size variable was set at 32 and the learning rate at 0.01.

## Evaluation

### Validation Dataset

For each image in the original dataset, 10 more images were augmented with random rotations and color/lighting/shadow effects. Altogether these make up the validation set which is used to evaluate the final model.

**Table 2.**

Class Labels	1	2	3	4	5	6	7	8	9
Sample Amount	726	759	847	726	836	1177	902	759	1023

### Performance - accuracy

For evaluation, the validation set was used to calculate the values found in Table 3.

Also using the inference results from the validation set, Table 4 takes into account only the true positive results in which the confidence was reported to be greater than 95%. The precision, recall, and F1-score is then calculated based on the True Positives being only correct predictions made with a confidence of 95%.

Precision and recall can help to determine how accurate the model is. The lower the precision value means the more the amount of false positives there are (Equation 1). As for recall, the lower the value, the higher the amount of false negatives (Equation 2).

The F1 is a function showing the balance between the precision and the recall. Where the function can be seen in Equation 3.

#### Equation 1. Precision

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} = \frac{\text{True Positive}}{\text{Total Predicted Positive}}$$

#### Equation 2. Recall

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} = \frac{\text{True Positive}}{\text{Total Actual Positive}}$$

#### Equation 3. F1-Score

$$F1 = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

**Table 3. Model Performance**

Class Labels	1	2	3	4	5	6	7	8	9
<b>True Positives</b>	723	747	847	687	833	1127	900	756	1003
<b>Total Predicted</b>	726	759	847	726	836	1177	902	759	1023
<b>Total True Label</b>	742	748	853	723	836	1170	901	771	1011
<b>Precision</b>	1.0	0.98	1.0	0.95	1.0	0.96	1.0	1.0	0.98
<b>Recall</b>	0.97	1.0	0.99	0.95	1.0	0.96	1.0	0.98	0.99
<b>F1 Score</b>	0.99	0.99	1.0	0.95	1.0	0.96	1.0	0.99	0.99

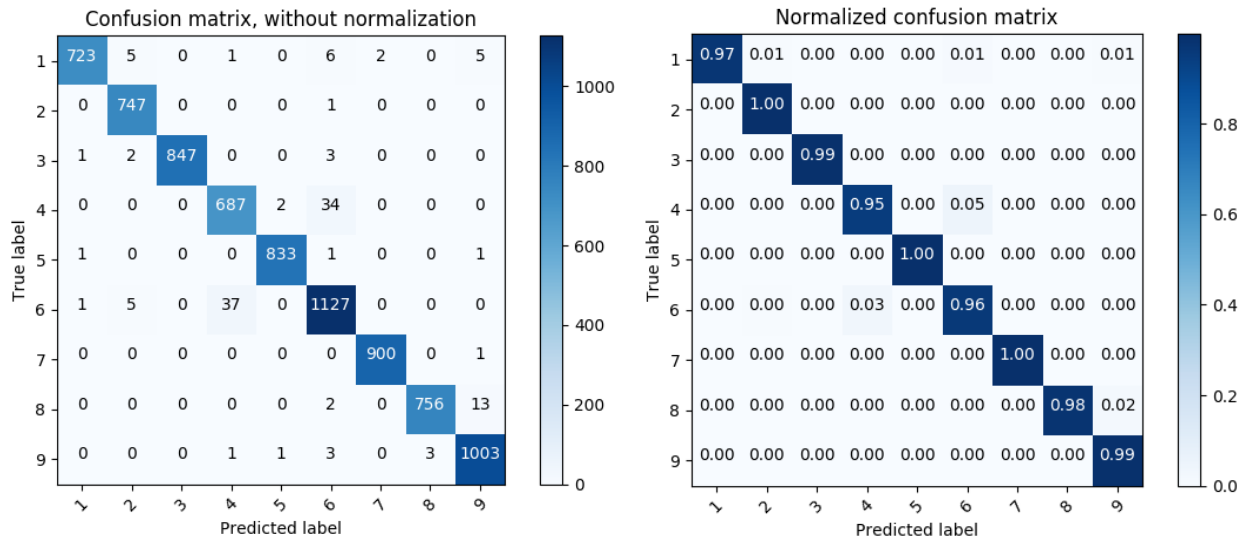
**Table 4. Over 95% Confidence**

Class Labels	1	2	3	4	5	6	7	8	9
<b>True Positives (CI &gt; 95%)</b>	607	573	742	263	646	445	772	587	797
<b>Total Predicted</b>	726	759	847	726	836	1177	902	759	1023
<b>Total True Label</b>	742	748	853	723	836	1170	901	771	1011
<b>Precision*</b>	0.84	0.75	0.88	0.36	0.77	0.38	0.86	0.77	0.78
<b>Recall*</b>	0.82	0.77	0.87	0.36	0.77	0.38	0.86	0.76	0.79
<b>F1 Score*</b>	0.83	0.76	0.87	0.36	0.77	0.38	0.86	0.77	0.78

\*Using TP amount where confidence is over 95%

Using the information generated in Table 3, confusion matrices are plotted in Figure 3 to help better understand the results.

**Figure 3. Confusion matrices**



### Performance – time

As for when the model is deployed on a machine with a NVIDIA GTX1080 GPU, the average time it takes for inference is around 1.3 seconds. The exception is for the very first image on start up around 10s, as the graph needs to be loaded.

To test the inference speed, a batch of 10 random images were given as input for the model. The time it takes for inference (coming to a prediction) for each image is logged. The mean value of the 10 slightly different times are taken for each trial. 10 trials were completed altogether. The results of this test can be seen in Table 5.

**Table 5. Inference times**

Trial #	1	2	3	4	5	6	7	8	9	10	Avg. Overall
Mean Inference Time (s)	1.315	1.324	1.281	1.280	1.282	1.307	1.291	1.294	1.306	1.314	1.283

## Discussion

For this business application, it is more costly to have a false positive than a false negative, as a false positive will mean that an incorrect object will be used, while a false negative will allow the object to not be used at all.

The most important value to look at for the model performance in this case is the recall value, as the higher the value, the more accurate the model is for that object. With accurate being how Both testing/training set and validation set is based on the original dataset. Thus there is an overlap of the images.

Generally, the model performs well with all the classes being correctly classified over 95% of the time. Even though some classes have a high recall, it is also possible that the recall rate for having a prediction confidence of over 95% is very low. For example, class label 4 has an overall recall of 0.95, but looking at the recall rate for if only true positives with a confidence of over 95% is taken into account, it is only 0.36. This means that even though it correctly predicted the objects, the confidence is not up to the desired 95%. A similar case is also found in class label 6. The comparison between the two recall values can be seen in Table 6.

**Table 6. Comparison of recall**

Class Labels	1	2	3	4	5	6	7	8	9
Recall (overall)	0.97	1.0	0.99	0.95	1.0	0.96	1.0	0.98	0.99
Recall (CI<96%)	0.82	0.77	0.87	0.36	0.77	0.38	0.86	0.76	0.79

## Recommendations

For better evaluation, make sure that the original images are removed from the augmentation sets. Also, use images collected from production environment for validation.

Based on this evaluation, look into training data, especially for class labels 4 and 6. Analyze how they could be more distinctive, feature-wise, and retrain the model with new training data.

Reconsider the threshold of confidence. Why should all predictions be at a 95% confidence? Is that actually a realistic threshold?