Discussion & Results

1. Object Classification

Once all the experiments previously described in the methodology were undertaken, a way of extracting the results from each of the models created had to be decided. In most studies, researchers opted to create a confusion matrix, where all the True Positives, True Negatives, False Positives and False Negative classifications would be noted. This deemed to be problematic because to identify such results, these had to be done manually by going through each image one by one. With a dataset of around four hundred and fifteen test images, this option would not be possible since it would take a very long time to do so, therefore this wasn’t possible.

Instead of a confusion Matrix it was opted to go for 3 tables representing each configuration where each table would show the number of “Actual Classes” which are the number of labelled food objects in all of the images and the number of “Predicted Classes” for each of the six food items. This would give a general idea of how the model performed because it would be possible to see if the model classified more objects, less or maybe even amount. These can be seen in the tables labelled “Configuration 1”, “Configuration 2” & “Configuration 3”.

Configuration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Coin | Ghaq Tal-Ghasel | Imqaret | Pastizz | Zalzett Malti | Gbejniet | Qassatat | Totals |
| **Actual** | 413 | 138 | 184 | 137 | 92 | 0 | 160 | 1124 |
| **Predicted** | 416 | 137 | 170 | 146 | 46 | 3 | 166 | 1084 |
| **Difference** | +3 | -1 | -14 | +9 | -46 | +3 | +6 | -40 |

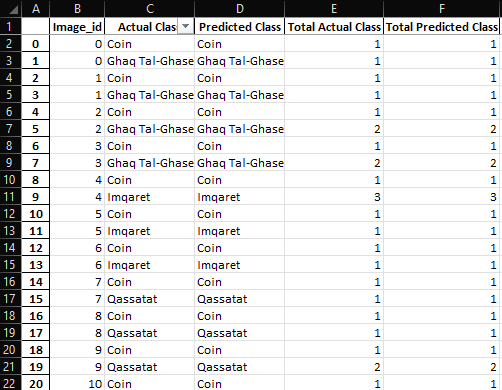
Configuration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Coin | Ghaq Tal-Ghasel | Imqaret | Pastizz | Zalzett Malti | Gbejniet | Qassatat | Totals |
| **Actual** | 413 | 69 | 138 | 114 | 138 | 92 | 183 | 1147 |
| **Predicted** | 433 | 78 | 192 | 140 | 145 | 95 | 183 | 1266 |
| **Difference** | +20 | +9 | +54 | +26 | +7 | +3 | 0 | +119 |

Configuration

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Coin | Ghaq Tal-Ghasel | Imqaret | Pastizz | Zalzett Malti | Gbejniet | Qassatat | Totals |
| **Actual** | 413 | 92 | 92 | 137 | 161 | 92 | 137 | 1124 |
| **Predicted** | 413 | 93 | 93 | 206 | 160 | 92 | 137 | 1194 |
| **Difference** | 0 | +1 | +1 | +69 | -1 | 0 | 0 | +70 |

These tables where generated by extracting the Actual classes from the mask found in the annotation files, whilst also getting the Predicted classes by looping through each image and trying to detect the items in each loop. These results where saved to a data frame so that it would be a bit easier to identify the classifications found in each image. Refer to *figure 1* for a visualization of these extracted results.



Figure

It can be noticed that from each of the configurations, the foods detected varied a lot between one another. There were two main reasons why this occurred. The first being that some objects where not even identified. In some cases, such as *figure 2*, food objects which where part of the six being used for this study, were present but not classified as one of the objects. As previously mentioned in the methodology section of this research, “fake objects” where also included as part of the dataset. This meant that there would be a chance that some of these fake objects would be falsely identified as one of the six focused items. One of these fake objects was a Green Apple, and this could be seen confusing the model, because when one of the images which had predicted an access amount of “Pastizzi” was visualised, it could be noticed that in most images where there is a green apple present, it was identified as a “Pastizz”. *Figure 3* shows an example of these images. This is the other main reason why the discrepancies between the Actual and Predicted values where present. The totals for these configurations where then added together and divide by three to get the one single value as an average of how well the models performed. The average of Actual Total objects was that of 1132 (1132.667, rounded upwards) and the average Predicted Total was 1181 (1181.333, rounded downwards). However, with this data it was still very difficult to understand how well the classification performed, but at least this gave an idea of which classes proved to be the most problematic to classify due to the difference in values. The values for “Qassatat”, “Gbejniet” and “Ghaq Tal-Ghasel” showed that these where most probably identified with the least number of problems encountered because they had the least difference in all three scenarios.

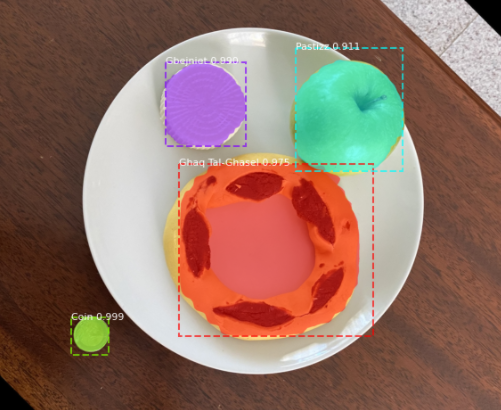
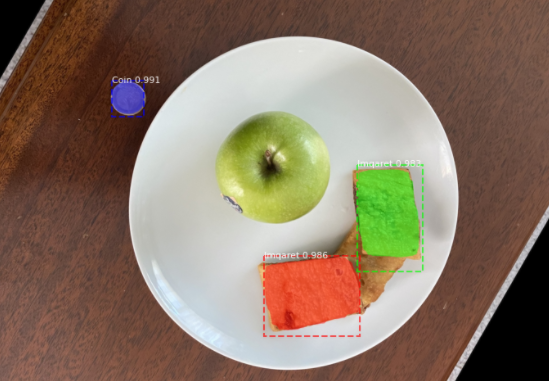


Figure 2

Figure 3

1. Calorie Estimation

Since the predicted classes in each image where identified, the next step was to extract information from the results, this helped in estimating the calories for each class detected. Previously in the Methodology section of this paper, it was stated that the neural network, “Mask R-CNN” was used to generate the models because it was capable of segmenting instances whilst providing the exact number of pixels detected. This value was mainly used to generate the nutritional value, because in previous sections, the pixels per cm2 and the calorie per cm2 were found, thus by using those values and the number of detected pixels it was possible to generate these results.

To get the best idea possible of how the model performed, multiple approaches had to be taken to make sure that a better understanding of what effects the final metric. The first approach which was taken included a simple comparison of the total pixels from each item’s mask with the predicted pixels value. Since each of the testing images had a mask included in the annotation file, it was simple to extract the mask’s total pixels value from the polygon and the predicted mask was also part of the results when detecting the food items from every image. The issue in this was that, if pixels were detected outside of the original mask, those pixels shouldn’t be counted, because of this it wasn’t possible to just compare the values together. Here it was opted to use the Intersect over Union (IoU) evaluation metric to get the accuracy of pixels detected whilst only considering the pixels correctly detected on the mask. So, by using this metric the accuracy of the model’s instance segmentation was determined. Configuration 1 yielded an IoU percentage of 87.12557%, configuration 2 resulted to 73.66437% and configuration 3 showed 80.5695% accuracy, therefore the average of all 3 configurations was that of 80.45315%. This value shows that model performed quite well since it managed to identify a very high percentage of pixels correctly, this was the first indication that there would be a greater chance for the estimation of calories to be accurate.

The next approach was to calculate the calories predicted based off the detected mask, then the same would be done for the actual calories but by using the original mask as the reference point. To calculate the calories, the reference object had to be used to scale up or scale down the predicted pixel values depending on the IoU value for the coin (the reference object in this case). Once this was calculated, this new value was divided by 3349.092, which is the previously found pixels per cm2, to find an estimation of the actual size of the detected object. The same procedure was applied to the actual value so that two values would be generated, the “Actual Kcal” and “Predicted Kcal”. To compare these values together, it was opted to use the evaluation metric of “Root Mean Square Error” to get a single value which would be able to represent the difference between the “Actual” and “Predicted” Kcal’s. The formula for this metric used was, . By applying this formula for each configuration, three RSME values where generated, 80.36378011, 79.30013 and 99.498. As previously done, these values where then averaged, which resulted in a final RMSE value of 86.38724. This meant that on average the models created would have a calorie estimate of plus or minus 86.38724. It is important to note that whenever multiple items from one food class are identified, the total calories for that food item are considered as a single aggregated value. For example, if two “Imqaret” where to be detected in a single image, only one pixel value for the total area of both would be extracted. The above information could be also visualized in *figure 4* which includes two small snippets of the results sheet to better understand the process of calculating these formulas.

The metrics which have been previously presented show give a very good idea how the model performed overall, but these calculations still had something missing, this being a comparison with the real calorie value. This was achieved by weighing the objects used in the dataset and manually calculating how much calories each item had, to see how the real value would compare with both the actual and the predicted calorie value. When looking at *figure 4*, it could be noticed that the “Actual Kcal” and “Real Kcal” do not match even though in theory these should match due to the calculations which where done. This happened because in certain cases objects have different weights, but their size might be very similar. Since throughout this research the calories where always calculated based on the cm2 of the object, if weight varies but the size doesn’t, the calculated “Actual Kcal” will not change, whilst the “Real Kcal” is based off manually weighing them, making it a more accurate value. The same steps where repeated to calculate the RMSE between “Real Kcal” and “Predicted Kcal”, which resulted in the three values of 94.19709, 138.2438, 75.49817 and an average of 102.6463. *Table 1* shows a table recap of all the three mentioned metrics used to show how the chosen approach performed in calorie estimation.

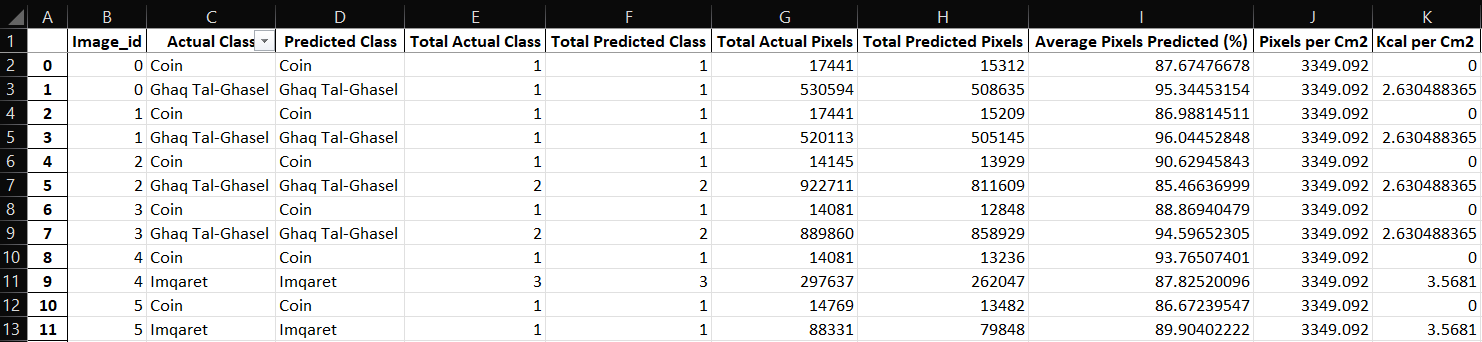
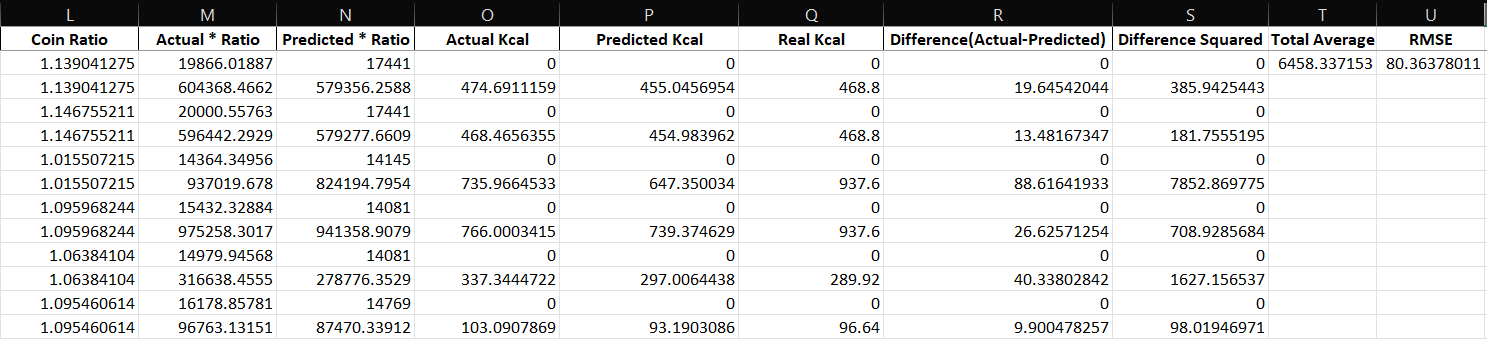


Figure 4

Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metric** | **Config1** | **Config2** | **Config3** | **Average** |
| IoU | 87.12557 | 73.66437 | 80.5695 | 80.45315 |
| RMSE (Actual - Predicted) | 80.36378 | 79.30013 | 99.49781 | 86.38724 |
| RMSE (Real – Predicted) | 94.19709 | 138.2438 | 75.49817 | 102.6463 |