DRP analysis model evaluation

By: Maurits Krijnen

Date: 20/06/22

Version: 1.1

Index

Preface

In this document I will be performing an analysis of the DRP detector machine learning model.

Here I will be showing the models current performance and explain the metrics I use to determine them. I also will be giving an advice at the end on how to proceed with the project in the future.

Model results

When evaluating the performance of a machine learning model we usually look at **evaluation metrics**. While there are many different evaluation metrics you can use, for this project I used the following: accuracy, precision, recall and the f1 score.

Statistics explanation

Here follows a short explanation of the 4 main statistics.

- **Accuracy:** or Classification Accuracy as it is also known. Is the ratio of number of correct predictions to the total number of input samples.
- **Precision:** is the number of correct positive results divided by the number of positive results predicted by the classifier.
- **Recall:** is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).
- **F1 score:** The F1 score can be interpreted as a harmonic mean of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

Evaluating performance

All of these metrics are scored on a percentage between 0-100%. In most cases scoring a higher percentile means more correctly guessed images, and a lower score means less correctly guessed.

Exceptions

This is not always the case however, for example a high accuracy % could mean that your data set is biased and requires cross validation. This could lead to various problems if not addressed. For this reason we combine accuracy with precision/recall/f1 score.

Model evaluation reports

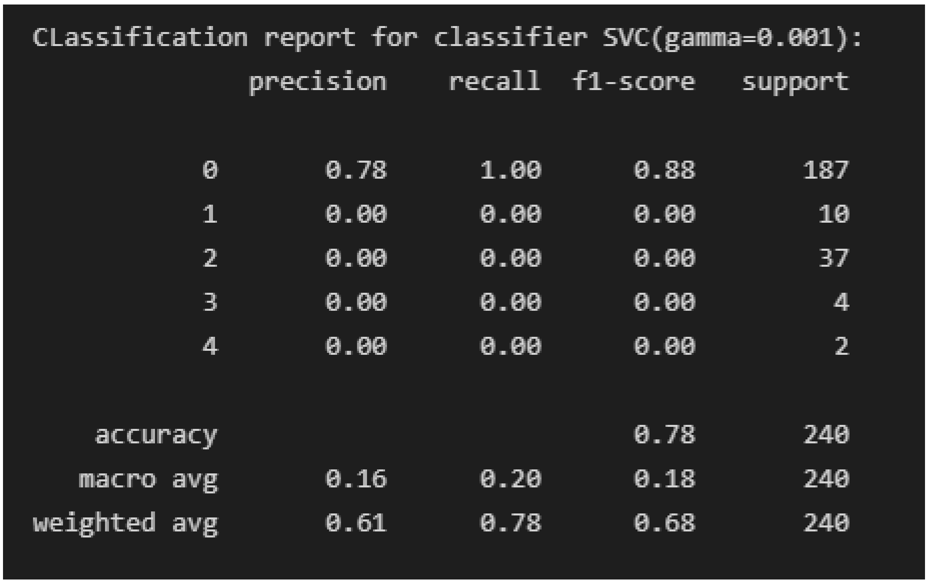
The following models were trained and tested using a data sample of 800 total images.
Note: Support stands for the number of images tested from the category listed under Classification.

SVC Model:

SVC made by taking random images.

As one can see this model scores 78% precision, 100% recall and 88% f1 score for category 0 and scores an average accuracy of 78%.

Initially this might seem like a good performance but as other categories are never chosen by the algorithm this means it is not functional and should not be trusted. A different way to look at it would be to say that the model would **never** predict that a user was sick regardless of the input image.



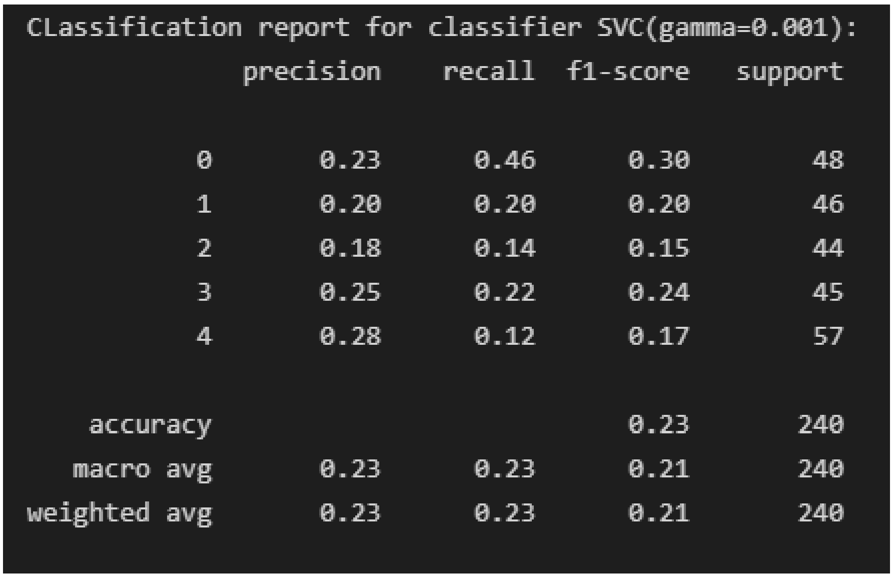
```
Classification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support

           0       0.78      1.00      0.88       187
           1       0.00      0.00      0.00        10
           2       0.00      0.00      0.00        37
           3       0.00      0.00      0.00         4
           4       0.00      0.00      0.00         2

    accuracy                           0.78       240
   macro avg       0.16      0.20      0.18       240
weighted avg       0.61      0.78      0.68       240
```

SVC Model with same sized data samples:

SVC made by taking the same amount of images for each category.

Here the performance is a precision of 18-28%, a recall of 12-46%, a f1 of 15-30% and an average of 23%. While this may seem like a much worse performance, this time the model is actually able to identify all categories similarly and identify both sick and healthy images. However the results itself are still lacking even if the model itself is fully functional.

**Image on next page*

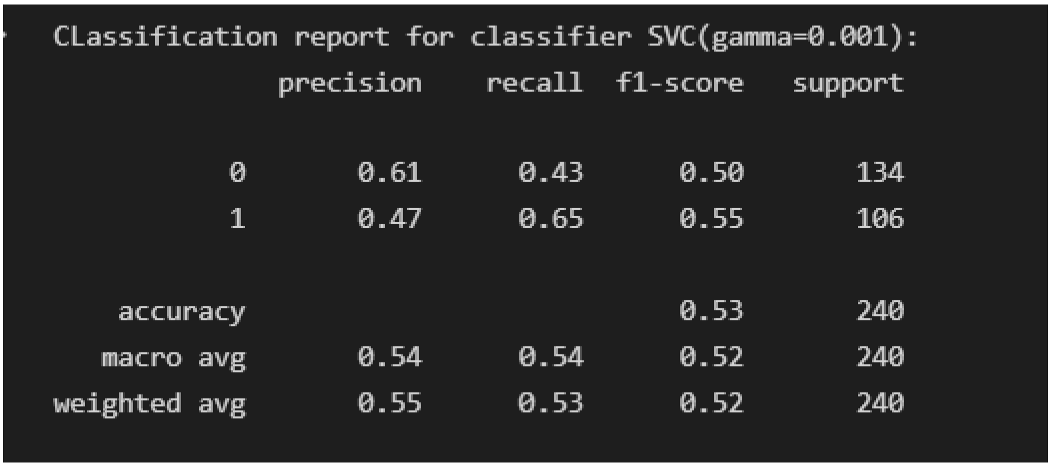


```
CLassification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support

           0       0.23      0.46      0.30        48
           1       0.20      0.20      0.20        46
           2       0.18      0.14      0.15        44
           3       0.25      0.22      0.24        45
           4       0.28      0.12      0.17        57

    accuracy                           0.23       240
   macro avg       0.23      0.23      0.21       240
weighted avg       0.23      0.23      0.21       240
```

SVC Model with same sized data samples and simplified categories:

SVC made by taking the same amount of images for each category but labeling 1-4 as 1s.

After consulting the stakeholder this idea was dropped as without the details of each category the model has no value to the project goal.



```
CLassification report for classifier SVC(gamma=0.001):
              precision    recall  f1-score   support

           0       0.61      0.43      0.50       134
           1       0.47      0.65      0.55       106

    accuracy                           0.53       240
   macro avg       0.54      0.54      0.52       240
weighted avg       0.55      0.53      0.52       240
```

Model Performance

What is the performance?

As you can see in the images above, the models are basically guessing. An accuracy of 53% for 2 categories or 23% for 5 just means the model is guessing and gets lucky a few times. This performance is somewhat disappointing as I was hoping for a score of at least 70% for the initial results.

Why is the performance so low?

There could be many different explanations as to why the score is this low. Low amount of data, low quality of data and the data is too similar to each other are a few that come to mind.
Next I'll go through my reasoning of why I believe these issues could potentially be present:

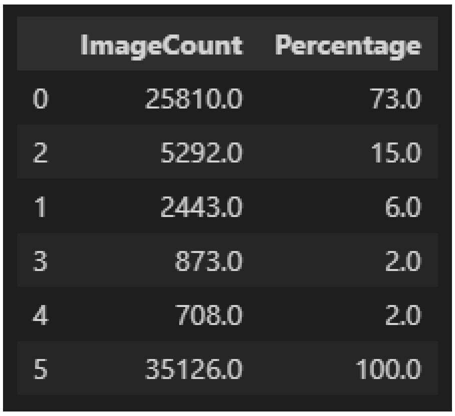
Low quality of data

While looking through the images I've started to notice that some images drastically differ in ways that are not relevant to the model's goal. For example other disease's, eye laser surgery or image quality distortions are present in at least 10-20% of images that I saw, particularly in category 4 which is the category with the lowest amount of data.

Low amount of data

While the amount of images needed per category for training may vary greatly between projects. AI research organizations mention[1] anywhere from 100 up to a few 100 are usually sufficient. For this project we have about 35000 images available so my initial thoughts were that we would have more then enough.

However after looking at the images per categories we have available this seems less promising:



	ImageCount	Percentage
0	25810.0	73.0
2	5292.0	15.0
1	2443.0	6.0
3	873.0	2.0
4	708.0	2.0
5	35126.0	100.0

As one can see, the majority of the data is category 0 and the lowest data count is category 4 with only 700 images. If we take into consideration the issue of data quality which is especially present in this category, we barely meet the recommended amount of data to train an average category. But in my personal opinion due to the high amount of variance inside each category, this project requires a model trained on more images then average.

source 1: https://www.folio3.ai/blog/how-many-images-are-required-for-deep-learning-classification/

Data(images) are too similar to each other

This issue is more speculative then based on proven evidence. When considering the symptoms of DRP, we're looking for internal bleeding near the smallest blood vessels in the eye. And in the case of proliferate DRP the growth of new small vessels.

Unfortunately category 0, 1 and 2 (or no DR, Mild and moderate) cases have very limited indication of the presence of the DRP. While an expert will know where to look the model will likely struggle identifying these factors

The reason for this could be that the low data quality combined with the relatively difficult to spot DRP indicators cause the model to train itself to identify other unrelated factors instead of the bleeding. This is hard to detect as I can't easily evaluate what area the model is considering for its judgment.

My advice

The nature of these issues lie with the dataset lacking in various areas. My suggestion would be to discontinue using this dataset and find a higher quality one either publicly available or request the stakeholder to provide one.

Alternative

If no other dataset is available, my advice would be to clean the dataset as much as possible by:

1 Detecting all faulty/incompatible images using some form of outlier detection

2 Generate more data by for example rotating existing high quality images

3 Edit the images in some for the Model to focus only on the bleeding

  - Examples: Highlighting the veins, zooming in on the area where bleeding is most common.

Conclusion

Looking at the various models I've created the performance is much lower then expected and required to achieve the project goal. There are various reasons why this is(or could be) the case and I've listed some suggestions on how to improve the model going forward with the project.

If this project does not finish within the allocated deadline a new project using a different dataset(but the same code to create the model) would likely have a better performance outcome without any changes having to be made.