Table of Contents

Formulation of the problem

Artificial datasets generated by models and algorithms can significantly reduce the effort required to solve computer vision tasks. We intend to test some of these methods for simple toy problem in order to find the direction of further work. For such toy problem, one can choose a classification task. If the model fails to classify synthetic images, then it is unlikely that such images can be used for harder problems such as segmentation and detection.

Our goal is to train classification model on generated dataset and test it on actual images of fruits.

We choose 5 classes from fruit dataset for classification: orange, peach, kiwi, tomato and banana. 1000 images from each class are considered as a test set. Some examples of images from fruit dataset are shown below:

Dataset generation

The methods used to generate images can be divided into two classes: classical and those using neural networks. One of the simplest but surprisingly efficient classic method is so called “copy-paste” method. It was shown that it achieves remarkable results for segmentation and detection tasks. It should work for classification as well. The idea is to take a few masks of real classified objects and place them to various backgrounds. Implementation details are given below.

Neural networks have achieved tremendous success in image generation in recent years. So there is a wide range of models to choose from. Unfortunately the most powerful generation models such as DALLE 2 or Midjourney are not open-source. So we fix on Stable-diffusion model that comprises relatively low-resources usage, high speed and good quality. Of course there is plenty of room to explore further for examples image2image models.

Copy-paste generation

Here we closely follow the method described in Roboflow guide. Firstly five masks of each class were manually cut from the fruit datasets images. We have used subset of Google image dataset that was collected by Roboflow as a background images. This subset consists of 9700 images which do not contain fruits. Random amount of fruit images (masks) have been placed in random locations of background images. We have also ensured that the foreground images do not overlap. 200 images of each class have been generated. Examples are given below:

Stable-diffusion

The main concern with stable diffusion model one has to take into account is the proper text prompt. For generating fruits the following prompt has been found to work well: "a photo of whole <fruits> on the table. top view" with negative prompt: "sliced, halved, cut" to avoid generating of sliced and cut fruits. However some amount of sliced fruits images are inevitable in the model output. Examples of images generated by stable-diffusion are given below:

Training

Pretrained on ImageNet ResNet18 has been chosen as a basic classifier. This model is rather lightweight but still should be able to solve simple classification tasks. Standard augmentations has been used during training (Random rotation, blur, horizontal and vertical flip). Learning rate has been chosen to be 1e-4 and OneCycleLr has been used as a scheduler. Each model has been trained for 5 and 10 epochs. Then we chose the best model. After training we calculate final metrics on our test set that consists of 5000 images (1000 per class) of initial fruit dataset.

We have also trained the model on images that was selected as masks for copy-paste generation method (5 images per class). So the entire dataset consisted of 25 images in this case. Metrics obtained for this “few-shot” regime can be considered as lower bound.

Results and conclusion

We have kept the number of images to be 200 in each class for all runs. To compare different datasets generation methods we consider 3 datasets: dataset fully generated by copy-paste method, dataset fully generated by stable-diffusion and hybrid dataset consisted of 100 images from copy-paste and 100 images from stable-diffusion for each class.

It is interesting to look at the metrics obtained on both the validation and test sets. Validation set consists of generated images so it has the same distribution as training data. But test set is a part of fruit dataset is likely to have another distribution.

Results for **validation** sets are given in the table below:

Metric / Dataset	**accuracy, %**	**f1-score, %**
5 imgs/class	60	50
200 imgs/class copy-paste	84.0	84.5
200 imgs/class stable-diffusion	**100.0**	**100.0**
100+100 imgs/class (copy-paste + stable-diffusion)	92.0	92.2

Results for **test** set are given in the table below:

Metric / Dataset	**accuracy, %**	**f1-score, %**
5 imgs/class	43	32
200 imgs/class copy-paste	**94.8**	**94.8**
200 imgs/class stable-diffusion	82.2	82.3
100+100 imgs/class (copy-paste + stable-diffusion)	94.0	94.0

Interestingly, the model is trained to classify perfectly the data generated by the stable-diffusion network. High metrics on validation mean that the distributions generated by the stable-diffusion are very narrow and the training dataset is exactly identical in its characteristics to the validation dataset. At the same time, the metrics on the test set are not catastrophically low which looks promising.

On the other hand, the model that trained on the data generated by the copy-paste method performs worse on validation than on the test. Apparently, this means that the distribution of the generated data is quite wide. And indeed all fruits in the test sample lie on the same surface.

In conclusion for these task copy-paste method performs better. But one should keep in mind that copy-paste uses mask that were taken from real data. The number of masks is quite low (5 per class) but this method nevertheless utilizes real data. Stable-diffusion has not seen any real data at all. Its performance is lower but not as dramatically as one may expect.