Object Recognition - Contextual Data Augmentation

Joaquim Marset Alsina

November 3, 2023

1 Introduction

The present report describes the results of the multi-label classification performance on the VOC Pascal dataset when performing contextual data augmentation in the training set. In section 2, we describe how we have performed the data augmentation, and in section 3, we explain the results of the different data augmentation experiments.

2 Contextual Data Augmentation

When we do not balance the training set, we consider a probability of 50% to perform data augmentation on an image. We always perform 3 experiments placing either 1, 3, or 5 objects per image. And when augmenting an image, we make twice as many attempts as the number of objects we want to place. This is to ensure as much as possible placing the desired number when we do not allow overlapping, and/or we do not rotate/scale, as some object may not fit, and we have to discard it.

When placing an object, we always compute a random position where it could fit. We have defined an object to overlap another if the Intersection over Union (IoU) of their bounding boxes is higher than 15%. If we allow overlap, we can place it without problems. If we do not allow it, we compute 30 random positions and compute the IoU of each position's box with the image boxes. If some position does not overlap with all the image's objects, we choose that position. If no position is valid, we discard that object and try another one.

When rotating an object, we consider a random angle between -20º and 20º. If we do not scale it later, we do not allow the rotation to change the bounding box size, so the object may end up cropped. If we scale it later, we permit the rotation to increase the box size, as we will ensure when scaling that the factor leaves the object inside the desired size of 224x224. Therefore, when scaling, we initially consider a random factor between 0.8 and 1.2, but if the object is outside the limits, we will always down-scale the image applying a factor between 0.5 and 0.8.

When we have to balance the training set, we compute a fraction of the *person*'s instances, and we randomly add objects of the other classes until each class has reached that desired fraction. Therefore, we compute the number of objects to place per image, and we randomly sample from the remaining classes to balance. In all experiments we have considered $\frac{1}{4}$, $\frac{1}{3}$, and $\frac{1}{2}$ of the *person*'s instances, and histograms have been generated with the final balancing. Given that *person* has many instances (i.e. around 4200), we do not balance up to that number as it would suppose too many objects per image, probably leading to pretty bad results.

3 Training Results

We have used *MobileNet-v2* as a backbone, given that we want to study the effects of performing augmentation rather than getting the best results. This model has fewer parameters than others commonly used (e.g. ResNet50), leading to less inference time, and the performance in these simple tasks is not that different.
The complete model adds a head composed of a dropout layer with a 20% probability and a dense layer with 20 units. It receives batches of size 32x224x224x3 and uses binary cross-entropy loss for multi-label classification.
It is first trained during 50 epochs, freezing the *MobileNet-v2* layers and using an *Adam* optimizer with a learning rate of $1e^{-3}$, and a learning rate decay of $1e^{-6}$. Later, to try to improve the performance (i.e. F1-score and AUC), we unfreeze the top layers of *MobileNet-v2* (i.e. from layer 100 on-wards), now using a lower learning rate of $1e^{-4}$ without decay.

Below we present the train and validation F1-score results of the different experiments sets (i.e. each row of the figure) we have performed. For space limitations we only discuss the F1-score results, as we consider it the most significant metric for this task:

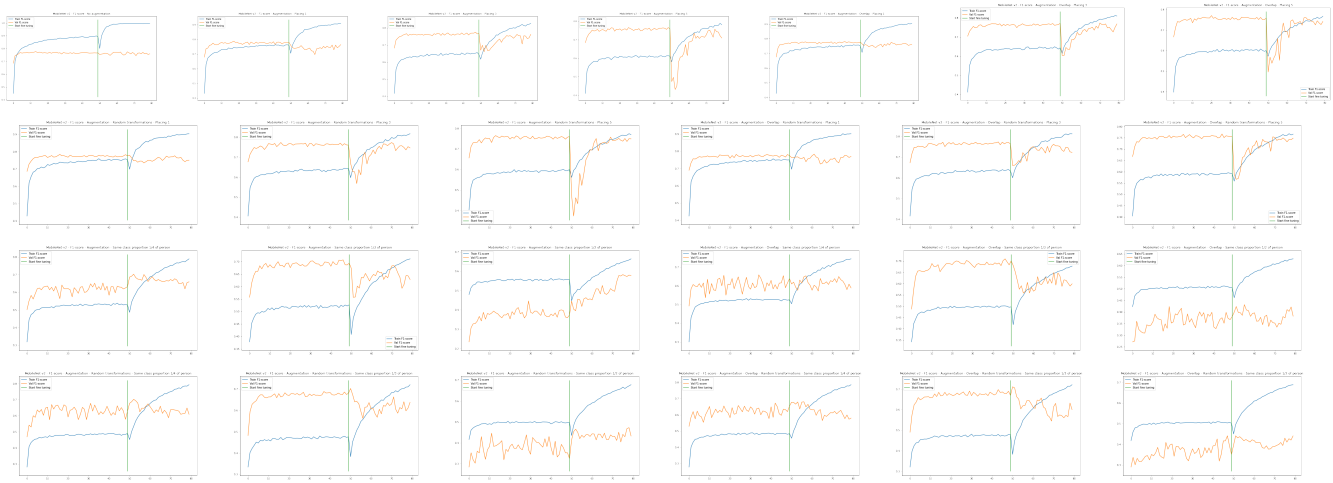

Figure 1: F1-score on training and validation sets during 80 epochs, 30 with fine-tuning

The first thing to note is that fine-tuning only helps in the training set and not in the validation, leading to some over-fitting. When the fine-tuning starts, the validation F1-score abruptly drops as the unfrozen layers are not stabilized with this data. We indeed reset the *Adam* weighted averages, but keeping them did not help either.

Applying augmentation, either with or without overlapping, do not suppose any improvement compared to not applying it. In the validation set, as more objects we place, the performance drop is higher when fine-tuning, but it can restore it, but never improve beyond. In the training set, with those 80 epochs, we see less over-fitting, but if we left more epochs, the gap would become as higher.

Then we can compare the results of applying random rotation and scaling. We have also tried either applying one of them, but we did not observe changes at all (these results are in the delivered folder). Therefore, if we consider both transformations, we again see that as fewer objects we place better results in the training set, but no change in the validation set. We again see the same pattern of performance dropping when starting the fine-tuning. Allowing the objects to overlap do not suppose any significant change in either the training or validation set when placing any number of objects.

When balancing the classes, we see a performance drop in both training and validation sets, with all the tried fractions, being the worst when we balance to $\frac{1}{2}$ and we allow overlapping. As expected, given that we add too many objects per image and a lot of occlusions will happen. Without the overlapping and the same fraction, fine-tuning improves the validation performance for the first time, reaching similar levels as the other fractions. Having overlapping or not in the other fractions do not suppose much change.

Finally, we have tried balancing the dataset and applying random transformations to the objects. Again, randomly rotating or scaling the objects do not seem to help at all, and worsens the results when balancing to $\frac{1}{2}$ and not allowing overlapping.

We can conclude that performing data augmentation has not helped at all in improving the model's generalization. Instead, it worsens the performance of the model, not making the effort, time, and resources worth it.
Neither has been performing random transformations. Even though we do not perform crazy transformations, we thought that letting the objects appear in poses and scales that the model may not have seen would help, but it seems that the already trained layers can handle them.
Balancing the training set worsens everything because we are biasing the object placing towards less-frequent classes. This will confuse the network as those objects will appear in backgrounds and around objects they should not, compared to randomly picking the object class to place each time.
Finally, overlapping does not improve anything, and even worsens a lot when balancing and placing a lot of objects, probably because of many occlusions, together with the drop balancing supposes in general.