DLCV HACKATHON REPORT

Abhipsa Basu

&

Ankit Kumar Gupta

Given to us were the pre and post-operative images of various people. The problem statement was to distinguish between pairs of images that belonged to the same person from those belonging to two different people.

The train folder consisted of two folders: pre and post, containing the pre and post-operative images of a thousand people. This meant that the labelled data had not been given to us, and hence we created it on our own.

Training Data generation:

This was an area of experiments where we created different distributions of the positive and negative samples and ran our model on them. Broadly we experimented on five ratios of positive:negative samples – 1:2, 1:3, 1:5, 1:10, 1:20. The positive sample generation was straight forward. We just matched the individual pre and post operative images of a single person to generate one positive sample. For the negative case, we randomly picked a sample from the pre folder, and a different sample (not of the same person) from the post folder. The number of such negative pairs depended on the ratio mentioned above. The ratio of 1:20 gave us the best results.

The dataset was then randomly shuffled, and then split into 75% training data and 25% validation data.

Model

Coming to the model, the idea was very similar to what is popularly known as the Siamese Networks. For each image in the image pair, we would:

- pass each of them onto a pretrained network (same for each image),
- obtain the last conv layer from each of those networks, converted them into feature vectors using GAP layer.
- So here we have obtained two feature vectors from the two images. We would simply concatenate them to obtain a single feature vector for both the images.
- Connect the concatenated layer to a fully connected layer of 1024 neurons (this was variable), with RELU activation
- Apply appropriate drop out on it and batch normalization
- Finally there would a classification layer of two neurons, predicting if the image pairs have "Same" person (1) or "Different" persons (2). The activation here would be softmax, as there are more than 1 neurons.

The most examined Learning rate was 0.001, and the categorical cross entropy loss was used with Adam optimizer.

Since the metric was macro f1 score, we used the same metric to monitor the training and validation performances.

One problem was that with this model, initially the predictions were all negative, as for 1 positive sample, there were 20 negative samples. We set the batch size to 128 so that each batch would see a good mixture of positive and negative samples, and also set the class_weights for the model.fit() method in keras. Class 1 would have a weight of 10, class 2 would have a weight of 1. This would mean that if a sample of class 1 were misclassified, it would be penalized 10 times more than the opposite case.

The best results were obtained using the pretrained ResNet 152 model, which gave a score of 0.70 on the public test data.