**The FaceNet approach**

FaceNet: A Unified Embedding for Face Recognition and Clustering.[2] FaceNet is a face recognition system developed in 2015 by researchers at Google that achieved the state-of-the-art results on a range of face recognition benchmark datasets (99.63% on the LFW). This work introduces the novel concept of triplet loss.

In this great article [6], Jason Brownlee describes how to develop a Face Recognition System Using FaceNet in Keras. Although the model used is heavy, its high accuracy is tempting to try using it. Also, as FaceNet is a very relevant work, there are available many very good implementations, as well as pre-trained models. Perhaps, by applying post-training quantization, the model could be reduced, and its speed would be good enough on mobile…

So, I decided to give it a chance and I converted David Sandberg’s FaceNet implementation to TensorFlow Lite. I’ve chosen this implementation because is very well done and has become a facto-standard for FaceNet. I thought that it was going to be an easy task, but I ran into several difficulties. I explain how I did it in this post.

Once I had my FaceNet model on TensorFlow Lite, I did some tests with Python to verify that it works. I took some images of faces, crop them out and computed their embeddings. The embedding’s matched their counterparts from the original models. I also noticed much lighter and faster execution with the Lite version on my laptop’s CPU.

As all of this was promising, I finally imported the Lite model in my Android Studio project to see what happened. What I found is that the model works fine, but it takes around 3.5 seconds to make the inference on my Google Pixel 3. The answers to the questions from the beginning, begin to be revealed.