We broke down the task in 3 steps:

1. Feature extraction from images
2. Triplet generation
3. Classification

For the first task we used a pretrained model: Inception-ResNet-v2, to obtain high quality features, which have great impact on the classifier performance. We also tried different nets like VGG16 or standard Resnet without the last fully connected layers but we settled for inception Net. The choice of the backbone, and consequently the features quality, greatly affected the overall performance of the net. Before extracting features we also performed normalization on the images to improve stability and performance.

After this we built triples of features, concatenating the extracted features from images as indicated on train\_triples.txt, and test\_triplets.txt for training and inference respectively.

Lastly we passed the triplets to our custom build classification net. You can see the structure in the code but basically it is a succession of dense layers, ending with a sigmoid activation to yield the binary output. To have some regularizing effect we added also dropout layers.

We utilized Adam optimizer with binary cross entropy loss.