To compute PCA, we need more samples than we have dimensions.

Use triplet loss for deep learning method. Lda for non deep learning. Embedding size approx equal to classes a good start. For triplet, use margin of one and normalise vectors. Could add augmentation to help curb overfitting by introducing more variance.

Is there a reason to convert the images to grayscale and resize them other than reducing compute?

For DCNNs, it's compute. For PCA/LDA you may want fewer dimensions do that number of samples > number of dimensions; and also computes an issue.

Somewhat obviously, colour is probably useful in this problem. If you reduce the resolution too much performance will plummet, but you can also go reduce this quite a bit without seeing a big drop. I think the images are 128x64 by default, going to 64x32 will still go pretty ok - starting to go below that though will see things struggle.

Hello! With regards to the question “are there any instances where the non-deep-learning method works better?” - what is meant by this (apart from that the runtime of a non-deep-learning method is much less)? Thank you!

There are 301 IDs. Overall you're probably seeing that the deep learning approach is better, but are there some IDs where the non-deep method works bette