Summary

	NN	NN-P	CNN	CNN-P	KNN	KNN-P
Accuracy	84.9%	94.3% (+ 9.4%)	97.7%	95.9% (- 1.8%)	84.0%	85.8% (+ 1.8%)

Legend:

<MODEL-NAME> - Default model without preproccessing

<MODEL-NAME-P> - Model with preproccessing applied to images

In this project, I decided to change the approach to the problem due to the nearly ideal image database. All the photos were in sufficiently high resolution, of the same size, and were taken on a green screen. I decided to use the ready-made database as the "modified" one, and I myself was modifying the images to make them more difficult to learn in order to test the algorithm's performance. With the default database, the results were very high. After changing the colors from green to black and from skin color to white - the algorithm achieved slightly worse results. It was visible that at times it had problems with classification, but it still maintained a level close to 90%. Of course, in the case of KNN and standard NN networks, I had to scale the images down to 30x30 pixels, as it makes learning for the algorithm much easier. They don't have the filters and the same scaling techniques such as CNN so giving them 150^2 inputs would be really inefficient. Despite this, in my opinion, the algorithms handled it really well. What's interesting, basic NN achieved a much better result with the preprocessing applied, and KNN achieved a slightly better result. At first I was surprised but then I remembered their resolution and it all makes sense. There isn't a lot of space for the color conversion function to make a mistake. It can even improve the algorithm performance.