CNN favours the bold – the untold story about seeing sign language in layers

Tristan Møller, Martin Jespersen & Marcus Gasberg
Aarhus University

Introduction

Ever felt the urge to understand sign language?

Stay a while and we'll tell you how a simple ML system can translate signs to words!

Data

Sign-language use visual manual modality to convey meaning. The system is fed pictures of signs made by a hand: 28x28px, 784 features

Try predicting this

Now imagine predicting hundreds of fast moving signs in real life ... not as easy, eh?

Taking the models for a spin

One does not simply choose one model, when investigating a ML-problem. It is necessary to try several, optimize them and find the best! All models are measured with a f1_micro score.

Logistic Regression Classifier – the 'benchmarker'

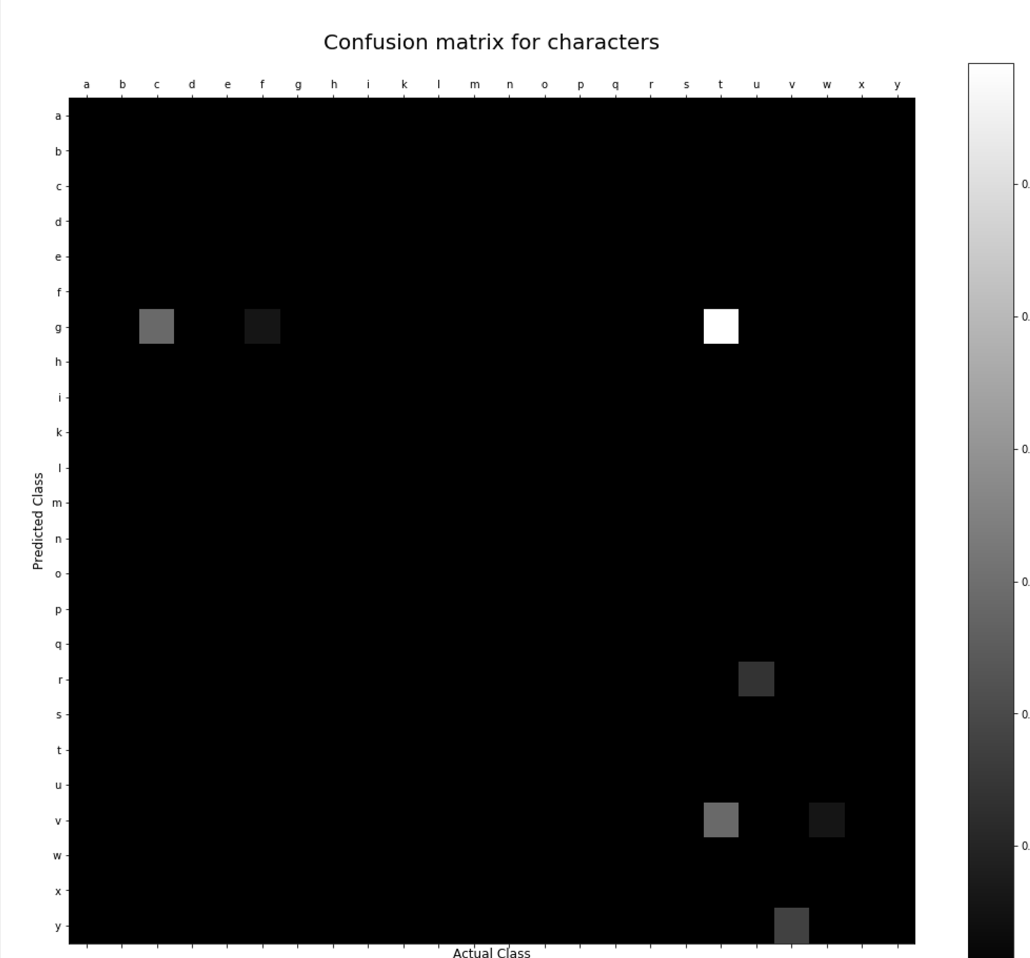
Scaling/normalization and GridSearch is used to find an optimal classifier. A solid score on 0.969. This model is unable to predict non-linear relations, which may cause the worse performance.

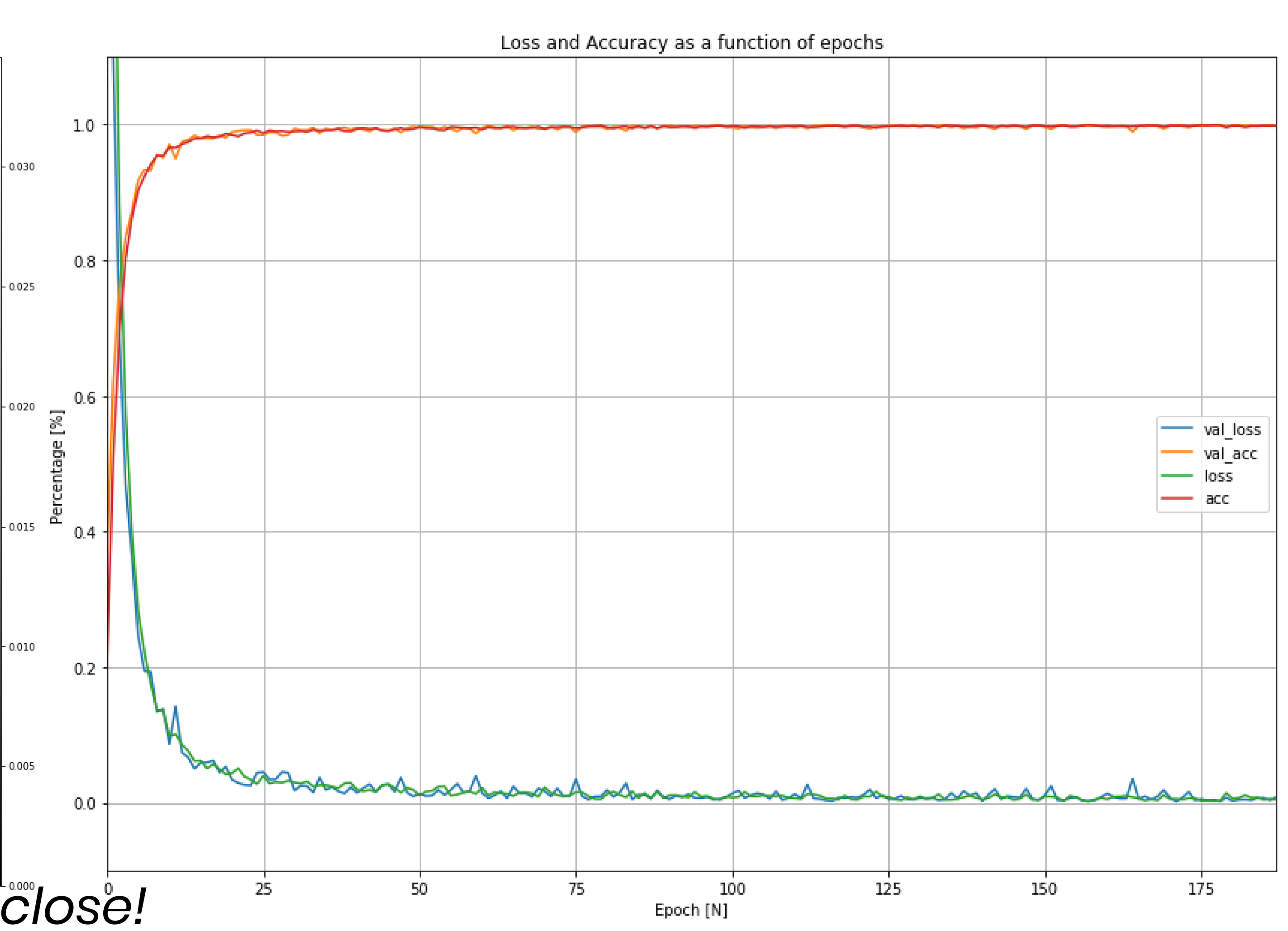
Support Vector Classifier – the smarter brother

Like before we use GridSearch to find a mean SVM classifier. It takes a few hours, but the result is a 'mean' model with a score of 0.986.

MMTNet (CNN Architecture) – The chosen one

A combination of data augmentation, convolutional/pooling layers & a 'mad' CNN architecture & voila, you have yourself a fully flexed ML system that can identify sign language with an impressive score of 0.996! Here is the performance of this bad boy:



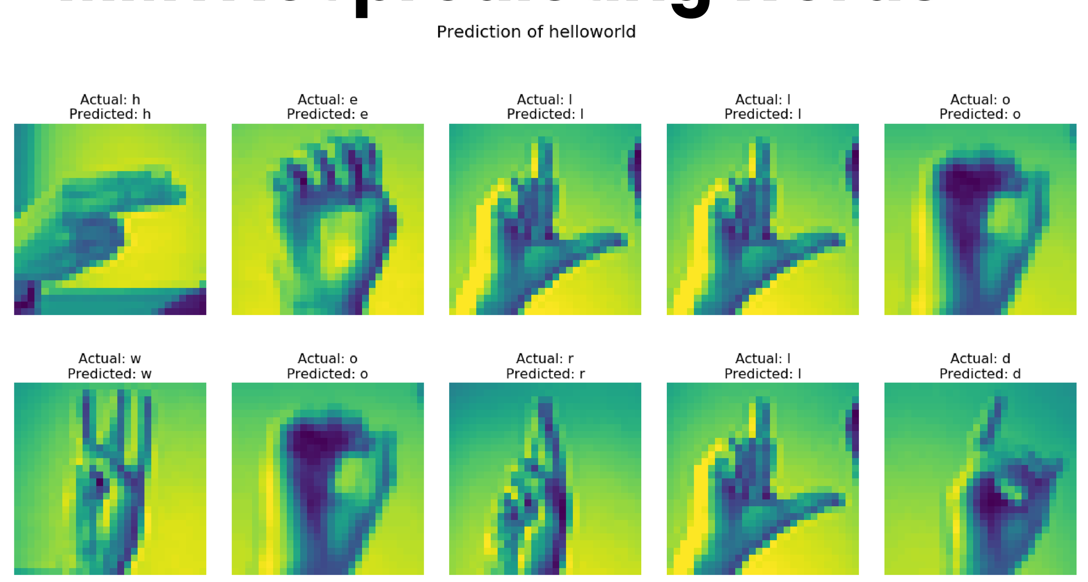


Now it ain't perfect, but it is quite close!

Data Augmentation

It turns out CNN requires a lot of data to avoid overfitting. It was necessary to use real-time generation of new signs when training the model. This increased the performance by 12%!

MMTNet predicting words



Who needs eyes when you have CNN?

Conclusion

CNN was designed for image recognition. Realistic data augmentation made the model less prone to overfitting. Convolution and pooling layers made it possible to detect higher-level features. These building blocks enabled the model to differentiate between signs and predict the letters.