A breakthrough in fingerspelling-to-text translation using CNN classification and Ngram modelling

Zhouxin Ge Bram Harleman Keith Klein Robert van Wijk

1. Contribution

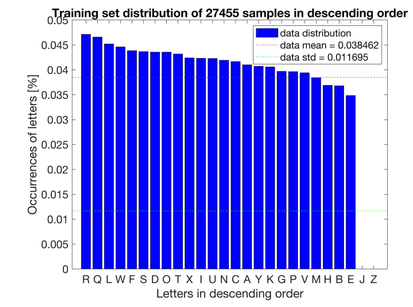
Learning sign language can be a difficult task. Fingerspelling, one of the sub-skills in sign language, is mainly useful for spelling names and places. This research contributes to a future application on fast and interactive learning of fingerspelling. In this work, a convolutional network is trained to classify sign language letters. An Ngram language model will assess whether the spelled letter is logical in the given context and alter the classification if necessary. This work proves flexibility of Ngram models to cope with a non-perfect classification to create better word predictions.

2. Hypothesis

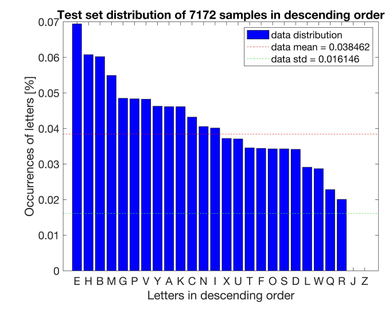
Convolutional neural network is a good, but not perfect way to classify letters being spelled. An Ngram language model can correct for these imperfections in the letter classification to be able to make a reliable prediction of the word that is being spelled.

3. Dataset

- CNN
  - MNIST American Sign Language Data-Set[1]
    - 27.455 (28x28px) training images
    - 7.172 (28x28px) test images
- Ngram
  - Brown corpus [3] used to calculate probabilities: over 1 million words
  - Google10000 dataset[2] used for testing

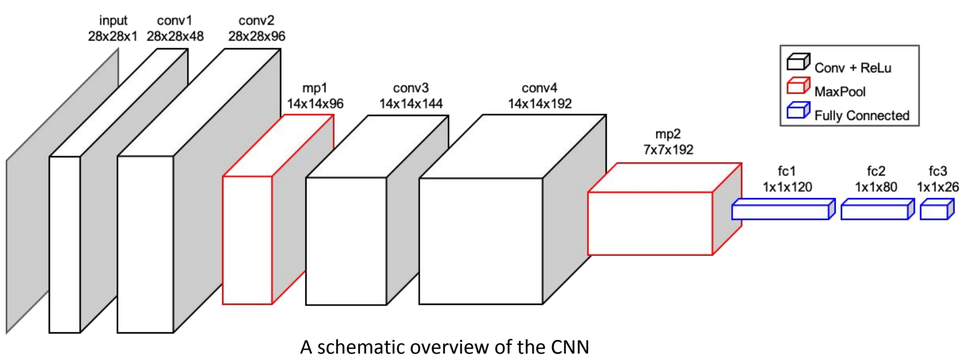


Portrayal of the ASL fingerspelling alphabet



4. Letter classification using a CNN

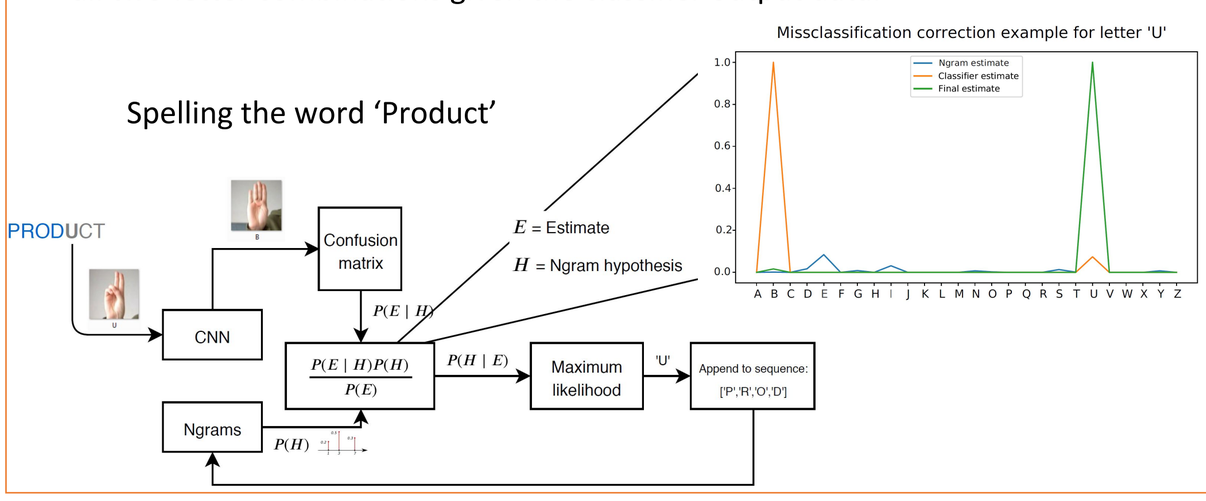
- Sign-language letter classification is done by a CNN.
- Architecture optimization by training 50 independent networks.
- The optimal architecture consists of two Conv+ReLu layers followed by MaxPool, this sequence is repeated once and concluded with three FC layers



A schematic overview of the CNN

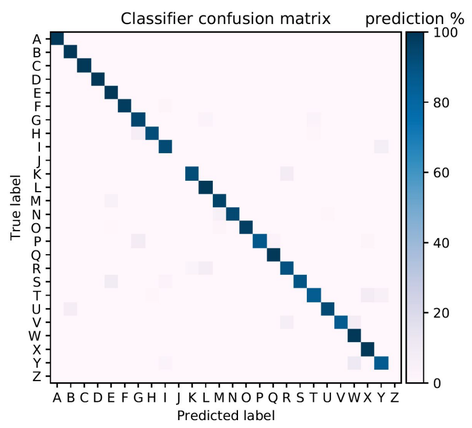
5. Adding Ngram

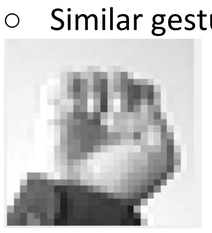
- Combining classifier probability distribution with Ngram hypotheses.
- Each letter is estimated by maximizing the posterior likelihood function.
- When possible, letters are estimated in pairs based on the maximum likelihood over all two-letter combinations given the classifier output data.



6. Results and Discussion

- 50 different CNN architectures were trained.
- 94.88% accuracy of best performing CNN
- Lowest class accuracy of 83% on letter R.
- Most trouble with letter combinations
  (True label, Predicted label):
  (R,L); (S,E); (U,B); (M,N);
- Probable reasons for confusion:
  - Low resolution images (28x28)
  - Similar gestures



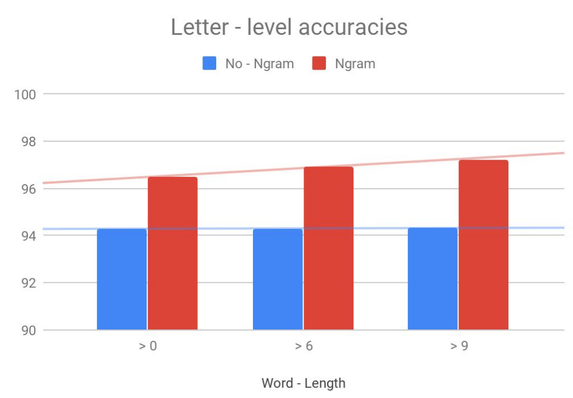


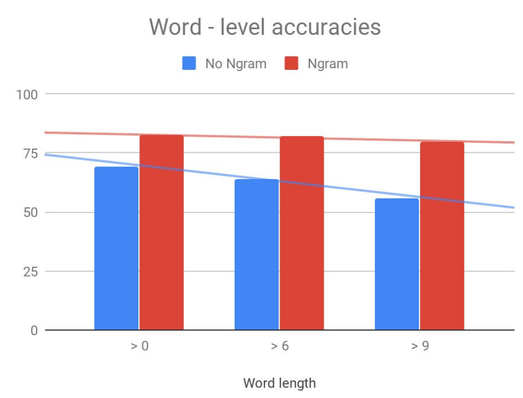
Letter E

Letter S

Letter M

Letter N





- The performance is compared on different minimum word-lengths.
- When adding the Ngram model, a performance increase is observed on the word-level as well as on the letter-level.
- Considerably better performance compared to classification-only, and less sensitive to the length of the word being spelled.
- Letter-level accuracy increases up to 97% depending on the length of the word.

7. Conclusion

A CNN with a simple architecture can be used to obtain acceptable performance on letter translation from ASL fingerspelling to text. However, the addition of a Ngram model can further increase translation performance, especially when spelling longer words. In the context of learning sign language, the predictive power of the Ngram can be harnessed to give feedback to the user input. The user can adjust his fingerspelling gestures to the given feedback, thus increasing the of speed of learning how to fingerspell.

8. Recommendations

- Investigate the difference in classification performance and speed when an LSTM is used instead of Ngram.
- Investigate if the found CNN performance is a local optimum.
- Add knowledge about letter index to Ngram model.

9. References

[1] Kaggle User: Tecperson (2017). Sign Language MNIST. Retrieved from https://www.kaggle.com/datamunge/sign-language-mnist
[2] Google's Trillion Word Corpus (2013), Github Repository retrieved from https://github.com/first20hours/google-10000-english
[3]Kaggle User: NLKT Data (2017). Brown-Corpus. Retrieved from https://www.kaggle.com/nltkdata/brown-corpus

TU Delft