Universiteit van Amsterdam

BSc KI, Project Thesis

# Modelling fonts with convolutional neural networks

Philip Bouman
10668667

supervised by
Dieuwke Hupkes, ILLC
Jelle Zuidema, ILLC

July 2, 2017

**Abstract**

We trained a convolutional neural network on the characters of 270 different writing systems in order to find relations between graphemes and phonemes. Using a phonological encoding scheme to represent phonemes features as vectors, the network was trained on images of graphemes to predict phonemes. Accuracy for predictive phoneme features ranged between 57% and 76% with an average of 63% for the best performing model.

# Contents

**1 Introduction** **2**
- 1.1 Bouba-Kiki effect . . . 2
- 1.2 Writing systems . . . 3
- 1.3 CNN . . . 3

**2 Method and approach** **3**
- 2.1 Evaluation . . . 3
- 2.2 Data . . . 4
- 2.3 Architecture . . . 6

**3 Experiments and Results** **6**
- 3.1 Accuracy . . . 7
- 3.2 Loss . . . 8
- 3.3 Loss total . . . 9

**4 Conclusion** **10**

**5 Discussion** **10**
- 5.1 Future work . . . 11

**6 Appendix** **11**
- 6.1 Writing systems used . . . 11

# 1 Introduction

Often in natural language there is no obvious relationship between the form and the meaning of a word. However in contrast to this arbitrariness there are instances of similarity between word form and meaning. A well studied phenomenon of such a relationship of iconicity is the *bouba-kiki* effect, in which shapes are consistently labeled with particular non-words. Iconic cross-sensory associations between sound properties and shape are generally accredited to be the source of this bias. Cuskley et al. [1] suggest that this phenomenon is also heavily mediated by the symbolic, culturally acquired shapes of letters. This similarity between orthography and abstract shapes forms the basis of this research, in which the aim is to find if this relation can also be found between graphemes and phonemes.

To gain a better insight to this entanglement of ortography and phonological features the goal of this research is to find if such relationships also exist between the shape and the sound of individual characters (letters). To avoid language-specific relations, a wide variety of writing systems (scripts) will be taken into consideration to discover general cross-cultural similarities between form and sound. To classify the sounds (phonemes) in a consistent way, a phonological encoding scheme will be used. This scheme contains all phonological features (whether a sound is nasal, labial, dorsal, etc.) of the phonemes collected in the International Phonetic Alphabet (IPA).

For this task a convolutional neural network will be trained to classify images of characters based on their related sounds. A deep convolutional neural network is a machine learning architecture consisting of multiple stacks of layers and has proved to be very succesfull for applications with two dimensional data. Machine learning also allows for a much wider range of writing systems to be analyzed than manually would be feasable. Furthermore this approach can provide an insight as to what specific properties of the characters are predictive phonological properties by investigating intermediate results between individual layers and smaller combinations of layers.

The results will be evaluated by measuring the level of accuracy on the test data.

# Background

The following sections provide some background information about the basis for this research, the data and chosen method.

## 1.1 Bouba-Kiki effect

The relation between word forms and their meanings in natural language is commonly referred to as sound symbolism or iconicity. A well studied phenomenon of sound symbolism is the *bouba-kiki* effect (also known as *takete-maluma*), in which participants label abstract shapes with non-word names. Shapes with

round curves are predominantly labelled with *bouba* and shapes with sharp edges with *kiki*. The effect occurs for children and adults [2] as well as in different languages [3]. The *bouba-kiki* effect is often ascribed to cross-sensory mappings between acoustic properties of sound and shape. The phenomenon might also be accredited to the symbolic, culturally acquired shapes of letters [1].

### 1.2 Writing systems

Graphemes represent the smallest individual characters of a writing system and phonemes relate to the sound of particular graphemes [4]. Phonemes represent a standardized collection of spoken language and are collected in the International Phonetic Alphabet (IPA) [5]. A phonological encoding scheme (1) is another way of representing the phonemes of characters. It is a more detailed description, containing the presence or absence of features in the pronounciation of a phoneme. Unlike the IPA there does not exist a standardized version of a phonological encoding scheme.

By using machine learning to classify graphemes to phonemes, a wide variety of writing systems[1] can be explored. These include Alphabets, Abjads (consonant alphabets) and Abugidas (symbols consisting of compositions of consonant and vowel).

### 1.3 CNN

The desired machine learnng architecture will be a convolutional neural network (CNN), since CNNs are among the most suitable architectures for handwritten digit and character recognition [6] and not yet used for this specific problem. Neural networks learn intermediate representations of their inputs, which can be useful for subsequent classification tasks. The structure of CNNs, mimicking receptive fields, is particularly useful for handling data that can be represented as two- or higher-dimensional input, such as images [7]. Deep CNNs also incorporate many layers and thus many intermediate representations, while keeping the number of free parameters small [8]. Recent work with CNNs on recognition of natural images and 3D objects resulted in higher accuracies than previously achieved by different methods. CNNs seem to work well for supervised and unsupervised learning tasks and can be trained on relatively small datasets [6].

## 2 Method and approach

### 2.1 Evaluation

The evaluation will be carried out by measuring the accuracy of the predicted phonemes for previously unseen graphemes using the test set. Depending on whether actual phoneme-grapheme relations exist, evalutation can be done on individual graphemes, simplified graphemes or grapheme clusters.

[1] Section 6.1

| IPA | consonantal | sonorant | continuant | delayed release | approximant | nasal | labial | rounded | strident | height | low | front | back | tense |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | - | + | + | 0 | + | - | - | - | 0 | - | + | - | - | 0 |
| æ | - | + | + | 0 | + | - | - | - | 0 | - | + | + | - | 0 |
| o | - | + | + | 0 | + | - | + | + | 0 | - | - | - | + | + |
| œ | - | + | + | 0 | + | - | + | + | 0 | - | - | + | - | - |
| e | - | + | + | 0 | + | - | - | - | 0 | - | - | + | - | + |
| ø | - | + | + | 0 | + | - | + | + | 0 | - | - | + | - | + |
| u | - | + | + | 0 | + | - | + | + | 0 | + | - | - | + | + |

Figure 1: Example Phonological encoding scheme

## 2.2 Data

All of the initial data was collected from Omniglot[2], an online encyclopedia of writing systems and languages. This data, consisting of 350 different writing systems, was trimmed down to a set of 275 different scripts (Section 6.1), due to either incomplete data or not being one of the categories (Abiguda, Abjad, Alphabet). For each language in the set, all the individual characters and their corresponding phoneme representations were extracted. These grapheme-phoneme pairs were then annotated by an expert to match IPA standards as well as to reduce the number of different phonemes (from 2000 to 117 distinct phonemes).

The phonemes in the data represent the pronounciation of the individual characters, thus alterations of sounds when used in combinations with other characters are not considered. Characters with multiple phoneme options are split and added for each possibility (E.g.:'پ', which can be pronounced 'b' or 'p', will be added as two distinct entries: 'پ', [b] and 'پ', [p]).

Lastly the phoneme-grapheme pairs are converted to their final form to be used in the CNN. For every grapheme a greyscale image of 32 x 64 pixels is generated. This is done by printing the charcater, using a language specific font, and converting it to an image file with the Python Image Library (PIL)(Figure 3). To achieve a consistent layout between the different characters Google NOTO fonts is used, which consists of separate OpenType (OTF) or TrueType (TTF) fonts for each (type of) writing system. Finally these images are converted to a matrix of greyscale values to be used as input for the CNN.

We chose to represent each phoneme/ IPA symbol as a vector with fourteen features with positive, negative or absent values (+, -, 0) (Figure 1 & 2). The following pronounciation features are considered: approximant, back, consonantal, continuant, delayed release, front, height, labial, low, nasal, rounded, sonorant, strident, tense.

From this data two different sets are created to be used in the model, both split three ways: training (80 %), validation (10 %) and test (10 %). The first dataset is split language wise, keeping characters of the same language in the

[2]http://www.omniglot.com/charts/#xls

| | |
|---|---|
| Кіь | [1, 0, 0, 0, 0, 0, 0, 2, 1, 1, 0, 1, 1] |
| Р | [1, 1, 1, 2, 1, 1, 0, 0, 0, 0, 2, 2, 2, 2] |
| Х | [1, 0, 1, 1, 0, 0, 0, 0, 2, 1, 1, 0, 0, 0] |
| Щ | [1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0] |
| Б | [1, 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 2, 2, 2, 2] |
| Ё | [0, 1, 1, 2, 1, 0, 0, 0, 2, 1, 1, 0, 1, 0] |
| Кв | [1, 0, 0, 0, 0, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0] |
| Л | [1, 1, 1, 2, 1, 0, 1, 0, 0, 0, 2, 2, 2, 2] |
| С | [1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2] |
| Хв | [1, 0, 1, 1, 0, 0, 0, 0, 1, 2, 1, 1, 0, 0, 0] |

Figure 2: Example Dataset

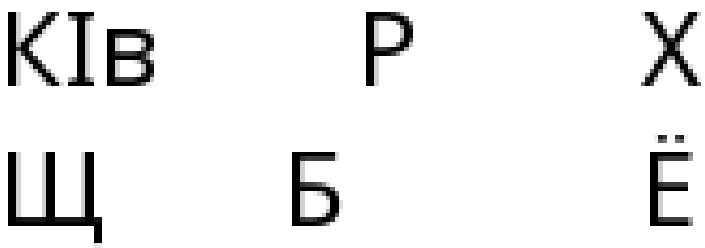


Figure 3: Character images

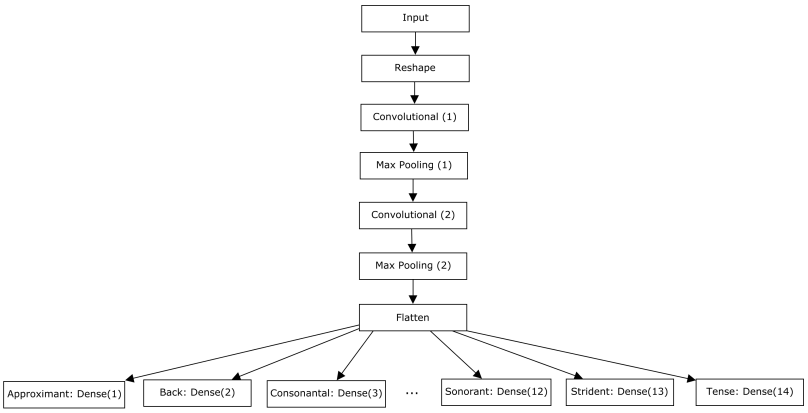


Figure 4: Model 1 Architecture

same set, the second set is split randomly on characters, shuffling the different languages. The choice for a second dataset was made to avoid any bias towards a particular writing system in the training phase.

### 2.3 Architecture

We trained two different models and applied them to the data. Both models were created with Keras [3] using the Theano [4] backend and the functional API class. The architecture of the first network (Figure 4) contains two convolutional layers and one fully-connected layer. Both convolutional layers have a kernel size of 3x3 and are followed by a Max-pooling layer. Then ReLU non-linearity is applied to the output of both convolutional layers. This architecture is similar to the network used on de MNIST dataset (handwritten characters) in LeCun et al. [9].

The second network (Figure 5) is similar to the first network, except for the addition of a third convolutional layer and a decreasing kernel size for each convolutional layer. The first convolutional layer has a kernel size of 9x9, the second layer kernel size is 6x6 and the final layer kernel size is 3x3. This architecture is similar to the approach taken in Krizhevsky et al. [7].

The input of both models consist of 32 x 64 pixel greyscale images of the characters and the output consists of fourteen classifiers for each pronounciation feature.

## 3 Experiments and Results

The two different models are used on the two datasets (language wise and random characters). Both training phases consist of 100 epochs with a batch size

[3] https://keras.io
[4] https://github.com/Theano/Theano

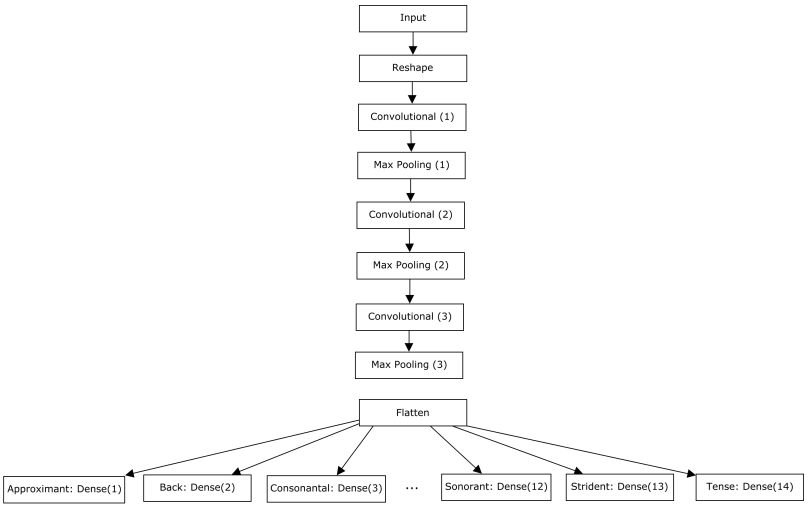


Figure 5: Model 2 Architecture

of 32. To compile the model the RMSprop adaptive learning rate method was chosen as optimizer with the default parameters (learning rate=0.001, rho=0.9, epsilon=1e-08 and decay=0). Sparse categorical crossentropy was chosen as loss function, which is the recommended loss function for multi-class models. The average run time for the first model was about 10 minutes, with around 5 seconds per epoch. The average run time for the second model was about 25 minutes, with around 14 seconds per epoch.

Model 1: two convolutional layers, fixed kernel size
Model 2: three convolutional layers, decreasing kernel size
dataset 1: split language wise
dataset 2: split character wise

## 3.1 Accuracy

The following table displays the average accuracy over 10 runs for each feature.

| | dataset 1 | | dataset 2 | |
|---|---|---|---|---|
| **feature** | **Model 1** | **Model 2** | **Model 1** | **Model 2** |
| approximant | 0.5934 | 0.6496 | 0.6216 | 0.6653 |
| back | 0.4567 | 0.5820 | 0.5337 | 0.5745 |
| consonantal | 0.5895 | 0.6815 | 0.5713 | 0.6835 |
| continuant | 0.6124 | 0.6392 | 0.6274 | 0.6696 |
| delayedrelease | 0.4823 | 0.5202 | 0.5451 | 0.5682 |
| front | 0.4832 | 0.5919 | 0.5285 | 0.5840 |
| height | 0.4832 | 0.5962 | 0.5309 | 0.5927 |
| labial | 0.9661 | 0.9659 | 0.8817 | 0.9797 |
| low | 0.4934 | 0.5835 | 0.5415 | 0.5672 |
| nasal | 0.7812 | 0.9738 | 0.7834 | 0.9769 |
| round | 0.5795 | 0.7222 | 0.6947 | 0.7570 |
| sonorant | 0.4379 | 0.5431 | 0.4808 | 0.6083 |
| strident | 0.5864 | 0.6614 | 0.6941 | 0.7039 |
| tense | 0.5059 | 0.5856 | 0.5342 | 0.5765 |

Overall model 2 achieved a higher accuracy than model 1 for both datasets. Furthermore dataset 2 achieved a slighlty better accuracy for most features than dataset 1. The accuracy for the labial and nasal features is very high and can be discarded due to the fact that almost all charcters have the same value (0, absent) for these two features. The highest scoring features after that for model 2 are round (0.7222 & 0.7570), consonantal (0.6815 & 0.6835) and strident (0.6614 & 0.7039) for dataset 1 and 2 respectively.

## 3.2 Loss

The following table displays the average loss over 10 runs for each feature.

| | dataset 1 | | dataset 2 | |
|---|---|---|---|---|
| **feature** | **Model 1** | **Model 2** | **Model 1** | **Model 2** |
| approximant | 2.6969 | 0.6343 | 2.1129 | 0.5978 |
| back | 4.3431 | 0.9466 | 2.6679 | 0.9200 |
| consonantal | 3.1131 | 0.6184 | 3.0880 | 0.5817 |
| continuant | 2.3414 | 0.6437 | 2.1179 | 0.6090 |
| delayedrelease | 3.2398 | 1.0105 | 2.3665 | 0.9467 |
| front | 3.8107 | 0.7964 | 2.9772 | 0.7984 |
| height | 3.7072 | 0.6560 | 2.9333 | 0.6182 |
| labial | 0.2680 | 0.1491 | 1.6994 | 0.0944 |
| low | 3.7548 | 0.9468 | 2.4025 | 0.9357 |
| nasal | 3.3102 | 0.1197 | 3.2707 | 0.0983 |
| round | 3.8070 | 0.5823 | 2.0709 | 0.5421 |
| sonorant | 3.7529 | 0.6596 | 3.7449 | 0.6207 |
| strident | 3.2295 | 0.7936 | 1.6814 | 0.7393 |
| tense | 3.5320 | 0.9144 | 2.5877 | 0.8941 |

Overall model 2 achieved a much lower loss than model 1 for both datasets.

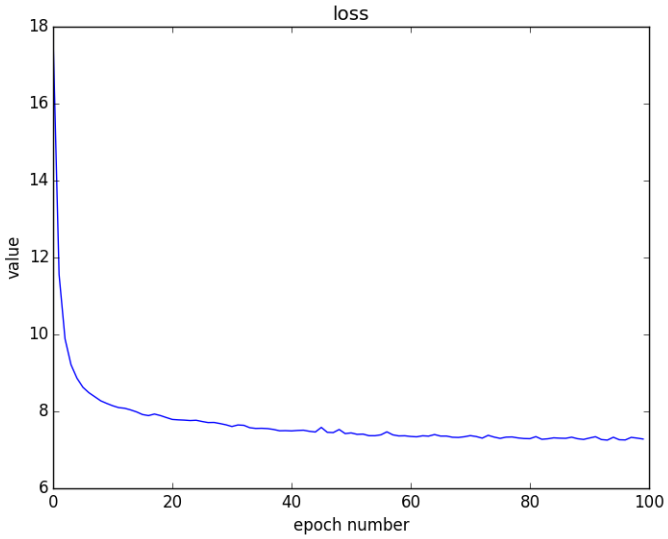


Figure 6: Loss total for one run

This is due to a few runs where model 1 did not converge (Figure 7). Furthermore dataset 2 achieved a lower loss overall compared to dataset 1. The loss for the labial and nasal is very low and can be discarded again due to the fact that almost all charcters have the same value (0, absent) for these two features. Apart from those two features the features with the lowest loss for model 2 are round (0.5823 & 0.5421), consonantal (0.6184 & 0.5817) and approximant (0.6343 & 0.5978) for dataset 1 and 2 respectively.

## 3.3 Loss total

The following table displays the mean and median for total loss over 10 runs.

| | **dataset 1** | | **dataset 2** | |
|---|---|---|---|---|
| **feature** | **Model 1** | **Model 2** | **Model 1** | **Model 2** |
| loss mean | 44.9066 | 9.4715 | 35.7212 | 8.9964 |
| loss median | 9.9079 | 9.9061 | 10.0021 | 10.0024 |

The model converged (Figure 6) for all runs with model 2, for model 1 however the model did not always converge resulting in a high loss average. Figure 7 displays the total loss for each run for both models and datasets.

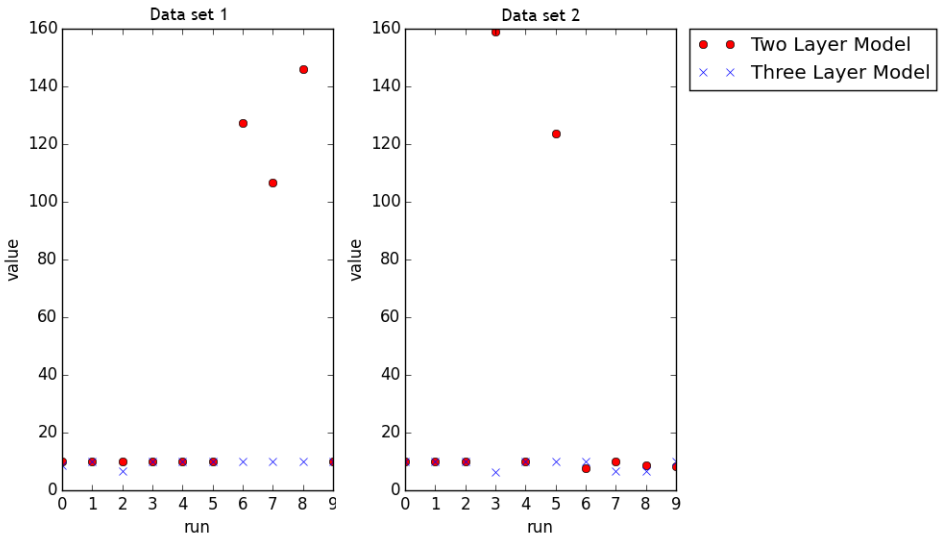


Figure 7: Loss total between different models

# 4 Conclusion

Evaluating the results shows that the model with three convolutional layers outperforms the model with two convolutional layers both on accuracy and loss. Furthermore the dataset with shuffled writing systems achieves better results than the dataset with languages kept together. The best predictive features in terms of accuracy are round, consonantal and strident. The best predictive features in terms of loss are round, consonantal and approximant. Making the features round and consonatal the best predictive feature on both evaluation metrics. The accuracy for the best performing model and dataset lies between 57% and 76% with an average of 63%.

# 5 Discussion

Based on the results no definitive claims can be made about the relation between graphemes and phonemes. Although some features show some promising predictive characteristics the overall acurracy is still quitte low. The features round and consonantal seem to be the most promising in predicting the correct phoneme. Better results are achieved when the dataset is split character wise in comparison with a dataset split language wise. The choice for a convolutional neural network as method for this task seems fit as improvements could be made by expanding the network.

### 5.1 Future work

To investigate the phoneme-grapheme relations further the following steps can be considered. First of all different or deeper architectures could be applied to this task. As shown extending the number of layers improved accuracy on the test set. Furthermore investigating the intermediate feature representations between layers or smaller combinations of layers could provide insights in predictive features of characters and their representations within the network. Using different or more phoneme features could also result in better performance. A typical phonological encoding scheme consist of many more features than used in this research and some features used proved not to be very informative (having the same value for most characters). Finally the type and number of writing systems used might lead to different results, as shown with the different results between datasets.

## 6 Appendix

### 6.1 Writing systems used

abaza, abenaki, abkhaz, acehnese, acheron, acholi, achuar-shiwiar, adamaua, adzera, afar, afrikaans, aghul, aguaruna, akan, akhvakh, aklan, akurio, alabama, albanian, alsatian, altay, alur, amahuaca, amarakaeri, andi, andoa, anuki, anutan, apache, arabela, arabic/cypriot, arabic/msa, arabic/tunisian, arabic/turkic, arakanese, araki, aranese, arapaho, arawakan, archi, are, arikara, armenian, aromanian, arvanitic, arwi, ashaninka, asheninka, assamese, asturian, atlantean, avar, avestan, avokaya, awing, aymara, aynu, azeri, babine, badaga, bagvalal, balti, bambara, bandial, basque, beaver, bedik, beja, bench, bengali, bhojpuri, bislama, bisu, bora, borgu, bosnian, bouyei, brahmi, brahui, burmese, burushaski, busa, bushi, caddo, capeverdeancreole, caquinte, carian, catalan, caucasian, cayuga, chapalaa, chavacano, chechen, chickasaw, chilcotin, chipewyan, chuukese, cofan, comorian, comox, coptic, cuneiform, cyrillic/finnougaric, cyrillic/other, cyrillic/romance, cyrillic/russian, cyrillic/slavic, cyrillic/tungusic, cyrillic/turkic, dagaare, danish, degxinag, delaware, dutch, dzongkha, eskimoaleut, estonian, ewondo, eyak, fijihindi, fula, futhorc, gitxsan, glagolitic, gothic, grantha, griko, guineabissaucreole, gujarati, hajong, hebrew, hindi, indonesian, interlingua, ipa, iranian, iroquoian, javanese, jeju, kabyle, kannada, karachaybalkar, karen, kashmiri, kayahli, kharosthi, khmer, khojki, khowar, korean, kove, kulitan, kumyk, kutchi, ladakhi, lao, latgalian, latin/aboriginal, latin/africa, latin/afroasiatic, latin/austroasiatic, latin/austronesian, latin/camerica, latin/celtic, latin/creoles, latin/english, latin/finnougaric, latin/formosan, latin/germanic, latin/hmongmien, latin/ial, latin/italic, latin/khoisan, latin/namerica, latin/nilosaharan, latin/samerica, latin/slavonic, latin/taikaidai, latin/tng, latin/turkic, latvian, lisu, lithuanian, lokoya, loma, lontara, lopit, lote, lycian, lydian, magahi, maithili, makonde, malay, malayalam, maltese, manchu, mandaic, mandarin, manipuri, marathi, marwari, maskelynes, mato, mayan, mende, mendekan, menominee, mongolic, monkhmer, mro, mutsun, nabataean, nadene, nepali, nheengatu, ocs,

okinawan, oriya, oromo, pali, pawnee, phagspa, philippine, pomoan, punjabi, quechuan, rarotongan, rejang, rohingya, romani, rovas, salishan, sankethi, sanskrit, shan, shina, siar, sikaiana, sikkimese, sinhala, sinitic, sio, somali, sundanese, sunuwar, sylheti, syriac, tami, tamil, teiwa, telugu, tengwar/arabic, tengwar/icelandic, tengwar/welsh, thai, tibetan, tokipona, tongan, tsakonian, tshangla, tulu, tuvaluan, ubykh, uto-aztecan, wandamen, westernrote, wichita, wolof, yabem, zigula

# References


[1] Christine Cuskley, Julia Simner, and Simon Kirby. Phonological and orthographic influences in the bouba–kiki effect. *Psychological Research*, 81(1):119–130, 2017.

[2] Daphne Maurer, Thanujeni Pathman, and Catherine J Mondloch. The shape of boubas: Sound–shape correspondences in toddlers and adults. *Developmental science*, 9(3):316–322, 2006.

[3] Robert D Tarte. Phonetic symbolism in adult native speakers of czech. *Language and Speech*, 17(1):87–94, 1974.

[4] Mark S Seidenberg. Beyond orthographic depth in reading: Equitable division of labor. *Advances in psychology*, 94:85–118, 1992.

[5] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[6] Dan Claudiu Ciresan, Ueli Meier, Luca Maria Gambardella, and Jurgen Schmidhuber. Convolutional neural network committees for handwritten character classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1135–1139. IEEE, 2011.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Théodore Bluche, Hermann Ney, and Christopher Kermorvant. Feature extraction with convolutional neural networks for handwritten word recognition. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 285–289. IEEE, 2013.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.