A ResNet-50-based Convolutional Neural Network Model for Language ID Identification from Speech Recordings

Giuseppe G. A. Celano
Leipzig University
Faculty of Mathematics and Computer Science
Institute of Computer Science
celano@informatik.uni-leipzig.de

Abstract

This paper describes the model built for the SIGTYP 2021 Shared Task aimed at identifying 18 typologically different languages from speech recordings. Mel-frequency cepstral coefficients derived from audio files are transformed into spectrograms, which are then fed into a ResNet-50-based CNN architecture. The final model achieved validation and test accuracies of 0.73 and 0.53, respectively.

1 Introduction

In the SIGTYP 2021 Shared Task, participants are asked to predict language IDs from speech recordings. The novelty of this Shared Task consists in (i) the variety of the languages involved, which comprises very different language genera/families (see Table 1), and (ii) the use of speech form.

Indeed, many linguistics-related Shared Tasks seem to focus on a restricted number of related languages (often Indo-European ones) and model their spellings.[1] In particular, this latter feature poses a number of theoretical and practical challenges, especially when some language comparison is involved, as in typological studies.

Writing systems, as is known, can highly diverge in what they represent, even when they are segmental scripts (not to mention that a language can be encoded in different writing systems, like, for example, Kabyle). If we consider the languages in the Shared Task dataset, it would be very hard to find a meaningful way to compare, for example, the Javanese writing system with the Portuguese one: the former could be written in the *scriptio continua* of its traditional script,[2] while the latter's alphabetical script distinguishes space-delimited tokens (mostly corresponding to morphosyntactic words). Interestingly enough, it is no less challenging to compare word-based scripts, in that there is no single definition of graphemic (let alone morphosyntactic) word across languages, and even within the same writing system, inconsistencies are not uncommon.

The use of language recordings instead of written documents should therefore ensure a more direct and consistent encoding of languages. Recordings also allow us to capture intonation structure, which is usually absent (or represented in a minimal form) in writing systems, despite its crucial role in conveying information (see Lambrecht, 1996 and, more in general, information structure studies).

On the downside, speech recordings are sensitive to idiolect variances, which a statistical model should however be able to properly address by not overfitting the training data. This is even more relevant for the SIGTYP 2021 Shared Task, in that its goal is to train a model being able to generalize to recordings of not only different people, but also very different genres/content.

In the following sections, I present the model I built to tackle the multiclass classification task at hand. In Section 2, the training and validation sets are described. Section 3 details the training phase of a number of models, including the ResNet-50-based CNN one, which I chose to participate in the SIGTYP 2021 Shared Task. Section 4 summarizes the results of the ResNet-50-based CNN model, while Section 5 contains some concluding remarks.

2 The training and validation sets

The training and validation sets are released by the organizers of the Shared Task as `npy` files containing mel-frequency cepstral coefficients (MFCCs) computed from audio files. The training set consists of 72,000 readings of the New Testament (each of them usually corresponding to a verse), while the validation set consists of 8,000 instances from different sources.

18 languages are included in the training set (4,000 instances per language), while only 16 lan-

[1] Interestingly, though, Gorman et al. (2020) concerns mapping of graphemes onto phonemes.

[2] Nowadays, however, Javanese is more commonly written in a Latin script.

Proceedings of the Third Workshop on Computational Typology and Multilingual NLP, pages 136–144

Language	Genus	Family	ID
Basque	Basque	Basque	eus
Eastern Bru	Katuic	Austro-Asiatic	bru
Hakha Chin	Gur	Niger-Congo	cnh
English	Germanic	Indo-European	eng
Hindi	Indic	Indo-European	hin
Iban	Malayo-Sumbawan	Austronesian	iba
Indonesian	Malayo-Sumbawan	Austronesian	ind
Javanese	Javanese	Austronesian	jav
Kabyle	Berber	Afro-Asiatic	kab
Kannada	Southern Dravidian	Dravidian	kan
Marathi	Indic	Indo-European	mar
Portuguese	Romance	Indo-European	por
Vlax Romani	Romani	Indo-European	rmy
Russian	Slavic	Indo-European	rus
Sundanese	Malayo-Sumbawan	Austronesian	sun
Tamil	Southern Dravidian	Dravidian	tam
Telegu	South-Central	Dravidian	tel
Thai	Kam-Tai	Tai-Kadai	tha

Table 1: Languages in the training dataset.

guages are in the validation set (500 instances per language, with the languages Eastern Bru and Vlax Romani missing). Each instance is encoded as a 2-dimensional tensor, whose shape is $(39, x)$, with $x \in \{x : x \in \mathbb{Z} \wedge 300 < x < 2729\}$.

MFCCs are often used as features in ML. Basically, they allow leverage of sound frequencies, which can offer a richer representation than that of a pure sound waveform (see Xu et al., 2004 for more details and their computation).

3 Method

3.1 A baseline model

A baseline can be calculated by feeding a model directly with MFFCs. The training and validation data contain tensors whose second dimension length varies. A solution for that can be slicing/padding them as to get shape $(39, 501)$, since about 80% of the training instances have a shape of $(39, x)$, with $x \in \{x : x \in \mathbb{Z} \wedge 300 < x < 502\}$.

A model is trained with three RNN layers and two densely connected layers, the last of which outputs the final probabilities for each label (see Appendix A). The RMSProp optimizer with learning rate 0.00001 is chosen. The first dimension of each input tensor can be interpreted as representing time steps or a sequence. Each time step (except the first one) receives the output of the previous time step:

$$h_t = tanh(Wx_t + Uh_{t-1} + b), \quad (1)$$

$$y_t = tanh(Vh_t + c). \quad (2)$$

At each time step, the relevant input vector x_t is multiplied by its weights and then added to the product of the (hidden) vector of the previous time step and its weights (b and c are the bias vectors, $tanh$ the activation function, and y_t the output vector).

The RNN model performs poorly (see Figure 1), since it cannot generalize at all. This is due not only to the model architecture, but also to the data mismatch between the sets, the validation data containing very different kinds of speech recordings. I therefore added part of the validation data (60%) to the training set and trained a new model with the same RNN architecture and hyperparameters. Figure 2 shows that this model returns very similar results: it also overfits the training data, the validation accuracy invariably remaining around 0.1.

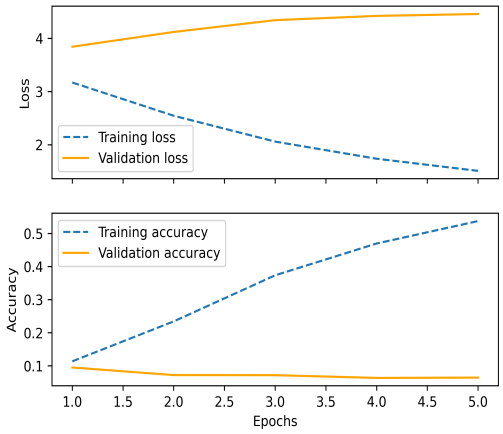


Figure 1: Performance of the baseline model.

3.2 A CNN approach

MFFCs can be used to create spectrograms, which allow transfer of a sound waveform into the image domain. Spectrograms return a visual representation of the unfolding of a sound wave through time, and have proved to provide promising results in a variety of ML tasks (see, for example, Chourasia et al., 2021 and Reddy et al., 2021).

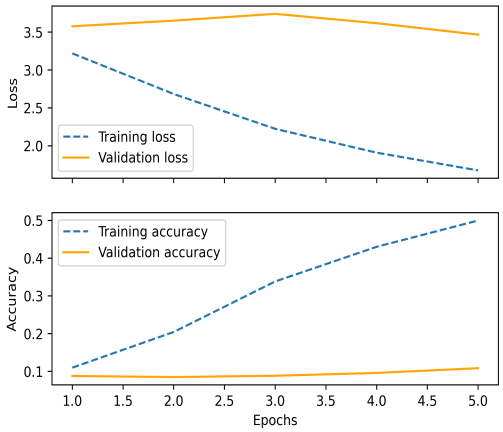


Figure 2: Performance of the baseline model with training set augmented with some validation data.

Using the default arguments of the function `specshow` (among which are `sr = 22050`, i.e., sample rate, and `hop_length = 512`) within the Python package `librosa`, the MFFCs are converted into images of shape $(640, 480)$ (Figure 3 shows an example of a spectrogram).

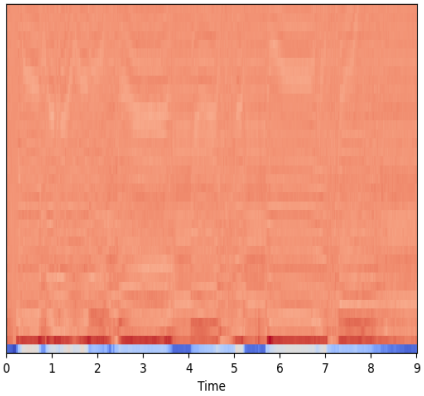


Figure 3: Spectrogram of a Hakha Chin instance.

The conversion allows one to take advantage of CNN architectures. In order to deal with the high variance of the model, 60% of the validation set is made part of the training set by stratified sampling: 300 instances of each language (i.e., 16×300) are randomly selected and added to the training set.

Two CNN architectures have been compared using the same dataset described above: a 3-layer CNN[3] and ResNet-50 (He et al. 2016). Despite its moderately deep architecture (see Figure B), the 3-layer CNN model (with RMSProp optimizer and learning rate 0.001) quickly overfits the training data (Figure 4) and therefore, like the RNN model, proves to be inadequate for the task at hand.

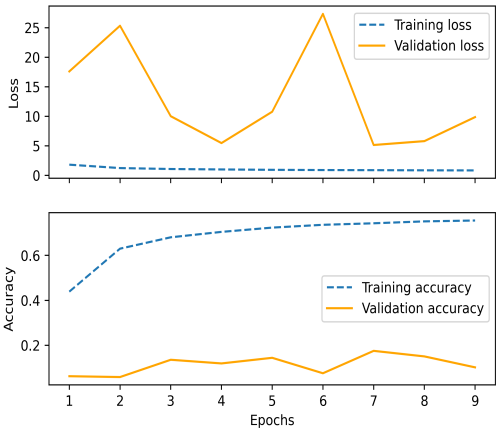


Figure 4: Performance of the 3-layer CNN model.

ResNet-50 is an extremely deep CNN architecture, which tries to overcome the degradation problem using residual learning. An input x is added to an output, so that a function $H(x)$ is redefined as

$$H(x) = F(x) + x, \tag{3}$$

which is hypothesized to make learning easier (He et al., 2016, p. 2). In Figure 5, one residual unit of ResNet-50 is shown: the layer `conv2_block1_out` is added to the layer `conv2_block2_3_bn` within the layer `conv2_block2_add`, as the same shape of the two layers shows $(120, 160, 256)$.

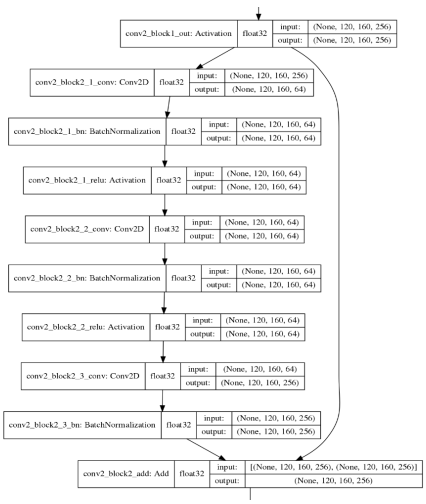


Figure 5: Detail of the ResNet-50 model.

There exist many ResNet architectures, such as ResNet-34, ResNet-50, and ResNet-101, each of

[3] 3 refers only to the CNN layers.

Language	Precision	Recall	F1
Eastern Bru	0.00	0.00	0.00
Hakha Chin	0.86	0.85	0.86
English	0.68	0.34	0.46
Basque	0.77	0.94	0.85
Hindi	0.92	0.68	0.78
Iban	0.95	1.00	0.98
Indonesian	0.78	0.64	0.70
Javanese	0.41	0.80	0.54
Kabyle	0.57	0.81	0.67
Kannada	0.92	0.73	0.82
Marathi	0.85	0.99	0.92
Portuguese	0.56	0.54	0.55
Vlax Romani	0.00	0.00	0.00
Russian	0.94	0.85	0.90
Sundanese	0.15	0.06	0.09
Tamil	0.84	0.77	0.80
Telegu	0.72	0.91	0.80
Thai	0.86	0.71	0.78

Table 2: Precision, recall, and F1 scores calculated on the validation set (ResNet-50-based model).

which is called after the number of the CNN layers and fully connected layers it contains. ResNet-50 has 50 of them, and according to the results reported by Xu et al. (2004), it performed better than ResNet-34, but worse than ResNet-101 and ResNet-134, in an ImageNet classification task (in reference to top-one and top-five error rates).

The ResNet-50 architecture has been employed to fit the training data of the SIGTYP 2021 Shared Task, without, however, transfer learning, in that the original weights were computed on completely different kind of data, and therefore are unlikely to be any useful. Of course, experimenting with different ResNet and non-ResNet architectures, as well as with different sets of hyperparameters, would be useful; the sizes of the architectures and the amount of training time needed to do that, however, made me focus only on ResNet-50, which turned out to return good results without requiring much optimization.

In order to accommodate the data of the SIGTYP 2021 Shared Task, the top layer was substituted with one allowing for the shape $(480, 640, 3)$, while the output layer was replaced by a densely connected layer outputting an 18-dimensional vector, i.e., a probability score for each of the 18 languages. The Adam optimizer with learning rates of 0.0001 (first 7 epochs) and 0.00001 (8th epoch) was chosen.

4 Results and Discussion

The ResNet-50-based model provides good training and validation accuracy scores (0.98 and 0.73, respectively). Importantly, both accuracy scores grow during training, and both loss scores get smaller and smaller. In Figure 6, the algorithm seems to have converged. However, the final accuracy score (0.53) calculated on the test set released by the organizers seems to suggest that some overfitting has occurred.

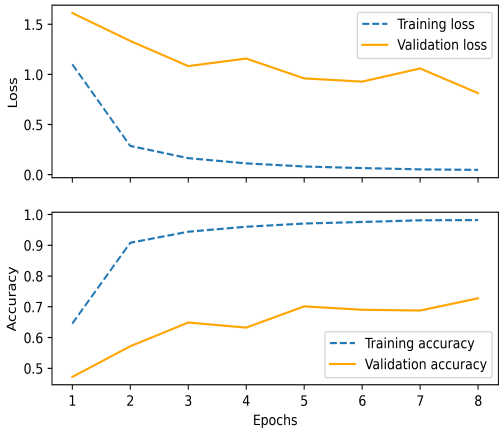


Figure 6: Performance of the ResNet-50-based CNN model.

The confusion matrices (Appendix C and D), the heatmaps (Appendix E and F), as well as the tables containing precision, recall, and F1 scores (Table 2 and 3), show that the model performs well, with a few exceptions. Sundanese is very often misclassified as Javanese. Appendix D reveals a more complex picture: English, Portuguese, Russian, and Thai are often also misclassified as Kabyle. Similarly, the model often associates Telegu with Kannada and Marathi. On the contrary, it can identify Iban very well. These results require further future investigation to ascertain whether these misclassifications can be ascribed to similarities between the languages.

Notably, the rows for Eastern Bru and Vlax Romani are not available in the heatmaps (Appendix E and F) because the languages are absent in both the validation and test sets.

Tweaking the hyperparameters and especially experimenting with deeper ResNet architectures could probably lead to an improvement of the results.

5 Conclusions

In the present paper, a ResNet-50-based CNN model has been presented, which was used to fit the data of the SIGTYP 2021 Shared Task. Attempts

Language	Precision	Recall	F1
Eastern Bru	0.00	0.00	0.00
Hakha Chin	0.72	0.81	0.76
English	0.48	0.37	0.41
Basque	0.69	0.93	0.79
Hindi	0.80	0.53	0.63
Iban	0.97	0.97	0.97
Indonesian	0.46	0.27	0.34
Javanese	0.41	0.83	0.55
Kabyle	0.36	0.06	0.45
Kannada	0.55	0.57	0.56
Marathi	0.57	0.51	0.54
Portuguese	0.31	0.43	0.36
Vlax Romani	0.00	0.00	0.00
Russian	0.33	0.04	0.06
Sundanese	0.21	0.10	0.14
Tamil	0.71	0.53	0.61
Telegu	0.44	0.73	0.55
Thai	0.64	0.29	0.40

Table 3: Precision, recall, and F1 scores calculated on the test set (ResNet-50-based model).

to tackle the task with relatively simple RNN and CNN architectures were unsuccessful. ResNet-50, however, proved to offer a robust architecture to train linguistic data for language ID prediction. The task at hand was challenging because the training data differ considerably from the validation data, and therefore any model needs strong ability to generalize. The ResNet-50-based CNN model proposed in this article shows good validation and test accuracies (0.73 and 0.53, respectively). Notably, Sudanese is very often misclassified as Javanese.

Acknowledgements

This work has been supported by the German Research Foundation (DFG project number 408121292).

References

Mayank Chourasia, Shriya Haral, Srushti Bhatkar, and Smita Kulkarni. 2021. Emotion recognition from speech signal using deep learning. In *Intelligent Data Communication Technologies and Internet of Things: Proceedings of ICICI 2020*, pages 471–481. Springer Singapore.

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya McCarthy, Shijie Wu, and Daniel You. 2020. The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50, Online. Association for Computational Linguistics.

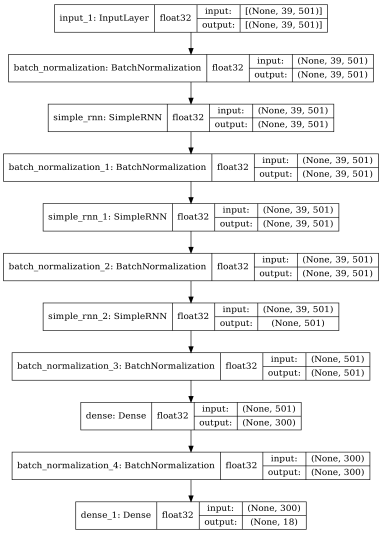
Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Knud Lambrecht. 1996. *Information Structure and Sentence Form: Topic, Focus, and the Mental Representations of Discourse Referents*. Cambridge University Press.

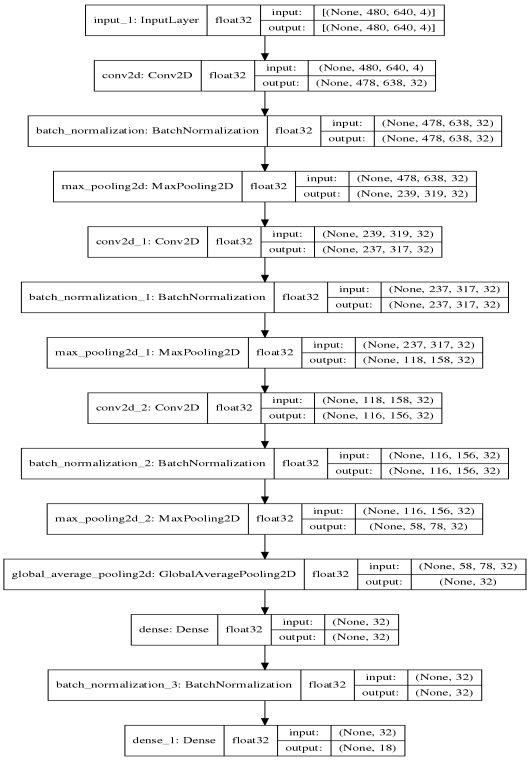
M. Kiran Reddy, Pyry Helkkula, Y. Madhu Keerthana, Kasimir Kaitue, Mikko Minkkinen, Heli Tolppanen, Tuomo Nieminen, and Paavo Alku. 2021. The automatic detection of heart failure using speech signals. *Computer Speech & Language*, 69:1–11.

Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. 2004. Hmm-based audio keyword generation. In *Pacific-Rim Conference on Multimedia*, pages 566–574. Springer.

A Architecture for the baseline model.



B Architecture for the 3-layer CNN model.



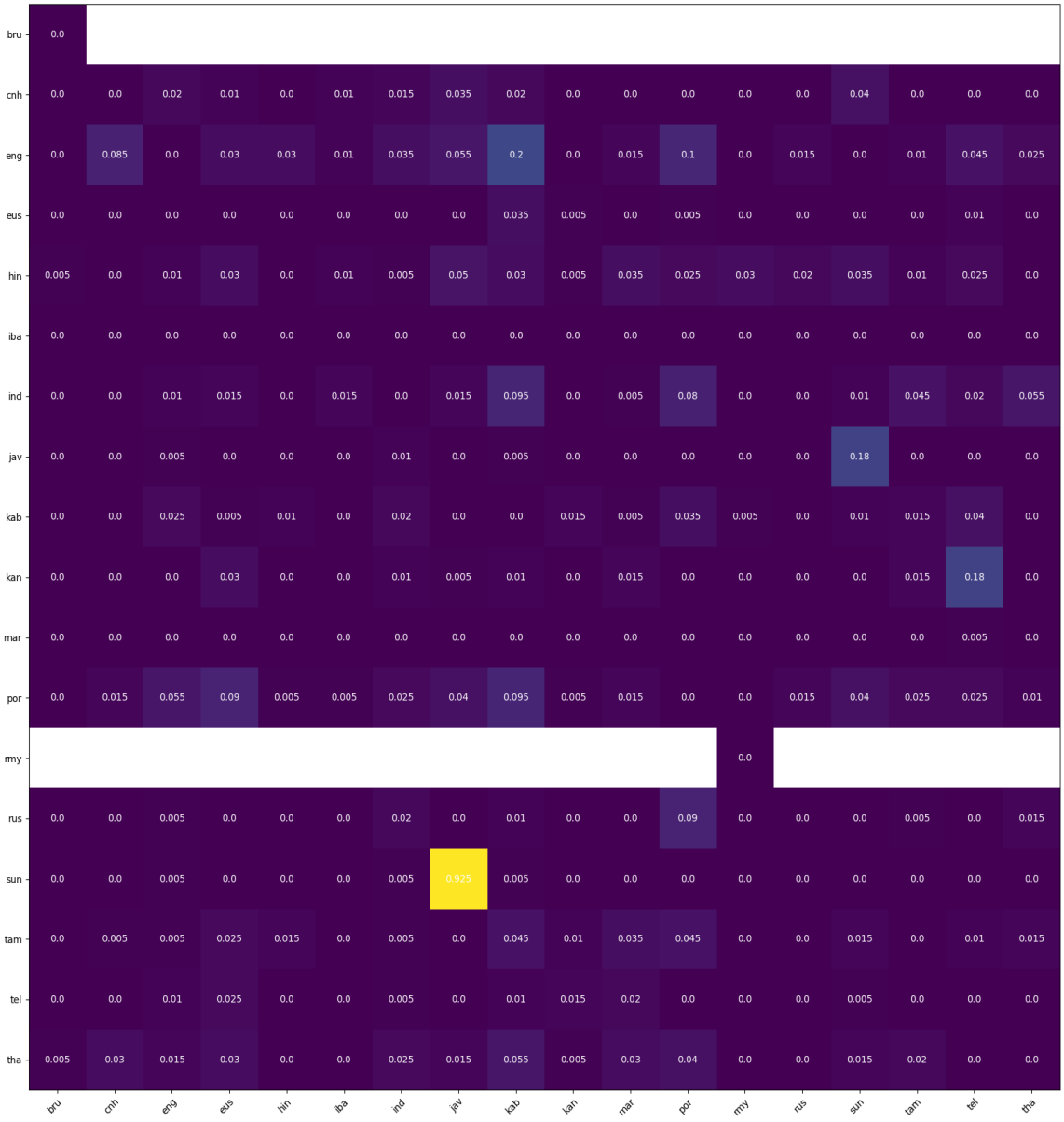
C Confusion matrix for the validation data (ResNet-50-based model).

	bru	cnh	eng	eus	hin	iba	ind	jav	kab	kan	mar	por	rmy	rus	sun	tam	tel	tha	class error rate
bru	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
cnh	0	170	4	2	0	2	3	7	4	0	0	0	0	0	8	0	0	0	0.15
eng	0	17	69	6	6	2	7	11	40	0	3	20	0	3	0	2	9	5	0.66
eus	0	0	0	189	0	0	0	0	7	1	0	1	0	0	0	0	2	0	0.06
hin	1	0	2	6	135	2	1	10	6	1	7	5	6	4	7	2	5	0	0.33
iba	0	0	0	0	0	200	0	0	0	0	0	0	0	0	0	0	0	0	0.00
ind	0	0	2	3	0	3	127	3	19	0	1	16	0	0	2	9	4	11	0.36
jav	0	0	1	0	0	0	2	160	1	0	0	0	0	0	36	0	0	0	0.20
kab	0	0	5	1	2	0	4	0	163	3	1	7	1	0	2	3	8	0	0.18
kan	0	0	0	6	0	0	2	1	2	147	3	0	0	0	0	3	36	0	0.27
mar	0	0	0	0	0	0	0	0	0	0	199	0	0	0	0	0	1	0	0.01
por	0	3	11	18	1	1	5	8	19	1	3	107	0	3	8	5	5	2	0.47
rmy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
rus	0	0	1	0	0	0	4	0	2	0	0	18	0	171	0	1	0	3	0.14
sun	0	0	1	0	0	0	1	185	1	0	0	0	0	0	12	0	0	0	0.94
tam	0	1	1	5	3	0	1	0	9	2	7	9	0	0	3	154	2	3	0.23
tel	0	0	2	5	0	0	1	0	2	3	4	0	0	0	1	0	182	0	0.09
tha	1	6	3	6	0	0	5	3	11	1	6	8	0	0	3	4	0	143	0.28

D Confusion matrix for the test data (ResNet-50-based model).

	bru	cnh	eng	eus	hin	iba	ind	jav	kab	kan	mar	por	rmy	rus	sun	tam	tel	tha	class error rate
bru	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
cnh	1	404	19	0	4	0	1	11	20	3	1	10	0	2	16	1	5	2	0.19
eng	1	72	183	8	3	2	9	9	105	4	9	45	0	8	9	5	20	8	0.63
eus	0	0	1	463	0	3	2	0	10	1	9	4	0	0	0	1	5	1	0.07
hin	4	2	9	14	264	6	2	65	12	6	39	17	16	2	31	2	9	0	0.47
iba	0	0	0	2	2	483	0	3	0	4	4	2	0	0	0	0	0	0	0.03
ind	1	8	2	47	4	2	134	32	51	7	4	63	0	2	26	40	21	56	0.73
jav	0	0	0	0	10	0	8	416	1	0	0	6	1	0	52	0	6	0	0.17
kab	0	3	44	17	13	0	11	11	300	7	2	51	0	3	7	10	19	2	0.40
kan	0	0	0	9	0	0	1	0	2	283	40	1	0	3	0	6	154	1	0.43
mar	0	0	0	1	0	0	0	1	4	88	257	0	0	0	0	0	149	0	0.49
por	0	13	47	18	2	0	17	16	95	2	7	215	1	14	6	21	22	4	0.57
rmy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
rus	0	3	23	48	17	2	27	0	84	36	0	225	0	18	0	4	11	2	0.96
sun	0	0	0	0	10	0	8	416	1	0	0	6	1	0	52	0	6	0	0.90
tam	0	20	4	9	3	0	21	20	46	15	27	19	1	1	21	267	20	6	0.47
tel	0	4	5	20	0	0	6	2	2	59	35	0	0	1	0	3	363	0	0.27
tha	0	30	47	13	0	0	47	18	93	1	19	37	0	1	22	17	9	146	0.71

E Heatmap with validation set error rates (ResNet-50-based model).



F Heatmap with test set error rates (ResNet-50-based model).