Language speech recognition

Aalto - ELEC-E5510

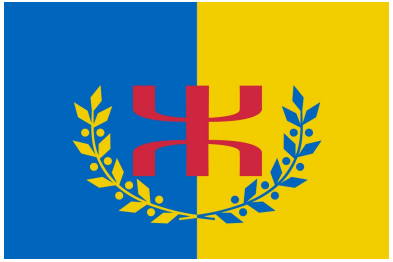
Álvaro Orgaz Expósito
Rachhek Shrestha

Audio samples from 6 languages

German - Mandarin - Spanish - Estonian - Kabyle - Farsi








Audio samples from 6 languages

Objective:

Determine language is being spoken in a speech sample.

Problem:

Humans recognize it through perceptual process inherent in auditory system, and the aim is to replicate human ability through computational means.

How to scientifically distinguish spoken languages to correctly classify speech samples?

Raw train and test audio data

From MP3 to WAV a6 16kHz normalized at -3dBFS

Subset of Mozilla Common Voice speech dataset (voice.mozilla.org)

Independent speakers in train and test splits

Raw train and test audio data

	Train			Test		
Language	**# Files**	**Total Hours**	**Avg Sec**	**# Files**	**Total Min**	**Exact Sec**
German	7053	9	4.59	494	24.7	3
Mandarin	4298	8.03	6.71	695	34.75	3
Spanish	6527	8.13	4.47	452	22.6	3
Farsi	6337	8.74	4.97	516	25.8	3
Kabyle	6676	8.36	4.51	486	24.3	3
Estonian	4818	8.91	6.66	487	24.35	3
Total	**35709**	**51.17**	**5.32**	**3130**	**156.5**	**3**

Features extraction

For each entire audio file with different durations:

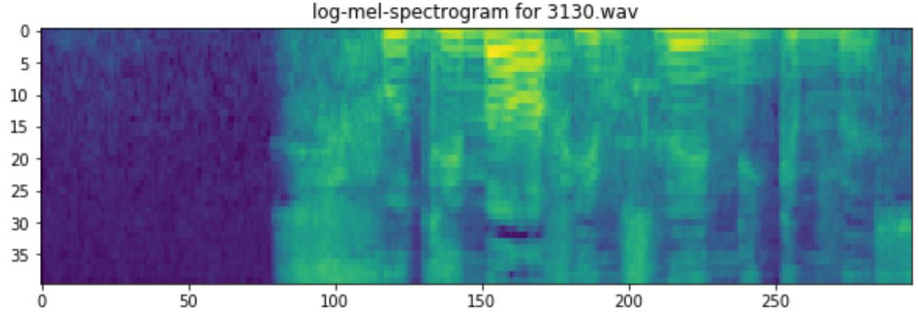
- Log-Mel-Spectrogram: 40 dimensions kept
- MFCC: 40 dimensions kept

Parameters:

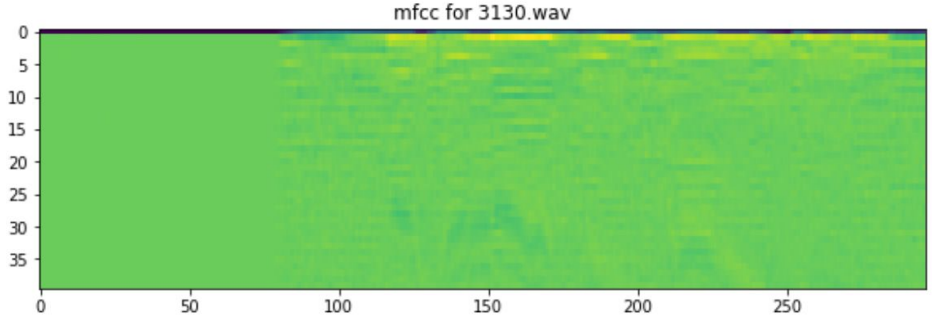
- N FFT = 512 (length of the FFT window)
- Hop Length = 160 (samples between successive frames)
- Frequency = 16000 (sampling rate)
- Fmin/Fmax = 300/8000 (lowest/highest frequency in Hz)

Features extraction

Example test audio file (exactly 3 seconds)



shape (297, 40) min -13.3
max 4.1 mean -4.1 std 2.5



shape (297, 40) min -648.6
max 191 mean -8.9 std 78.6

Preprocesing before modelling

Train files have different durations But we will **predict test files of 3”**

→ Model Input Frames = 297 = 1+(48000-512)//160

Split train features by moving windows of Hop Length = 100 frames

→ Equivalent to 100*160=16k samples or 1 second

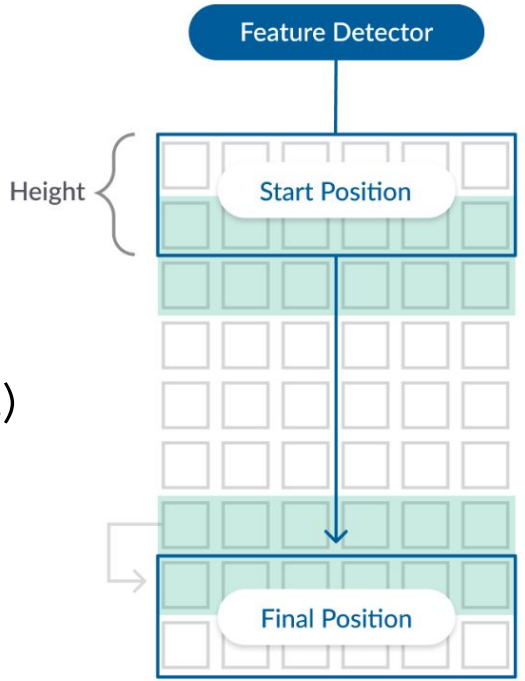
Why?

Consider time context, data augmentation (95k VS 47k samples from 35k files), and do not discard many remaining frames after last possible split.

CNN - Conv1D

model = Sequential()

model.add(Conv1D(filters, kernel_size, strides)



CNN - Architecture

Input (297, 40) - ReLU - Strides (1,2,1,1) - Kernel (6,7,1,1) - Softmax

Layer (type)	Output Shape	Param #
conv1d_1 (Conv1D)	(None, 292, 500)	120500
conv1d_2 (Conv1D)	(None, 143, 500)	1750500
conv1d_3 (Conv1D)	(None, 143, 500)	250500
conv1d_4 (Conv1D)	(None, 143, 3000)	1503000
global_average_pooling1d_1 (	(None, 3000)	0
dense_1 (Dense)	(None, 1500)	4501500
dense_2 (Dense)	(None, 600)	900600
dense_3 (Dense)	(None, 6)	3606

Total params: 9,030,206

CNN - Training configuration

Training with Keras Fit Generator (3GB-18GB) train data

Random mini batches → Each batch all languages

Batch size = 40 samples arrays of (297, 40)

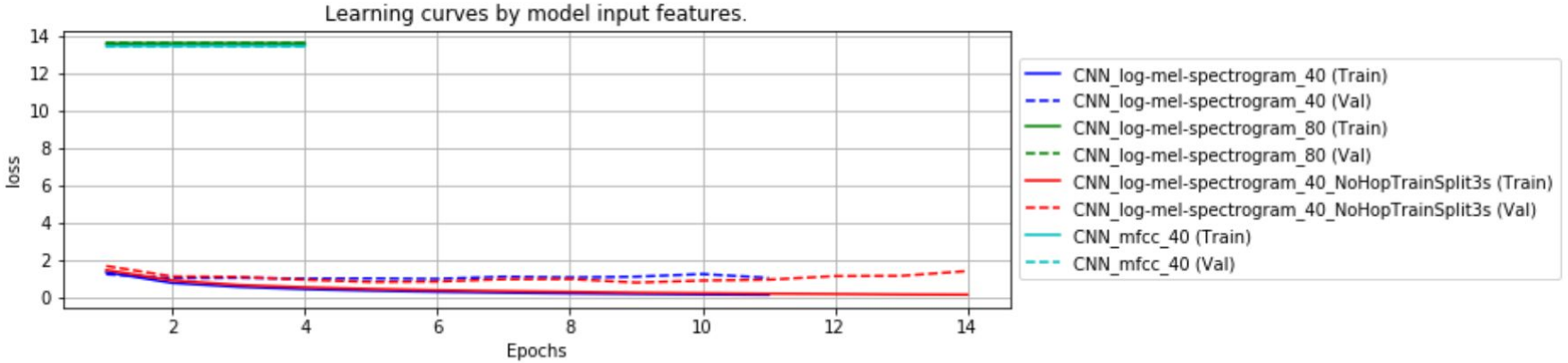
Each epoch = All training samples 95k

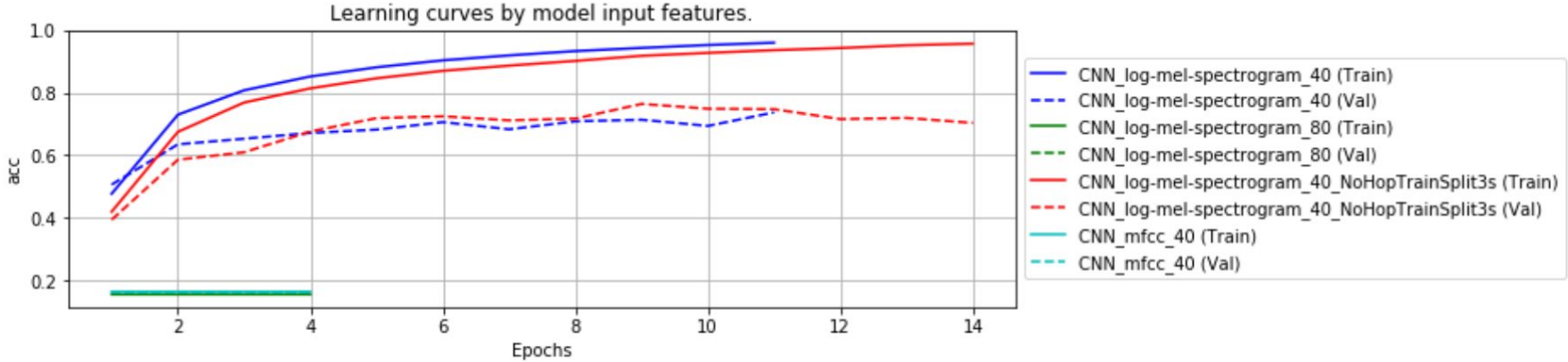
Loss Categorical Cross Entropy

Optimizer Adam Amsgrad LR=0.001

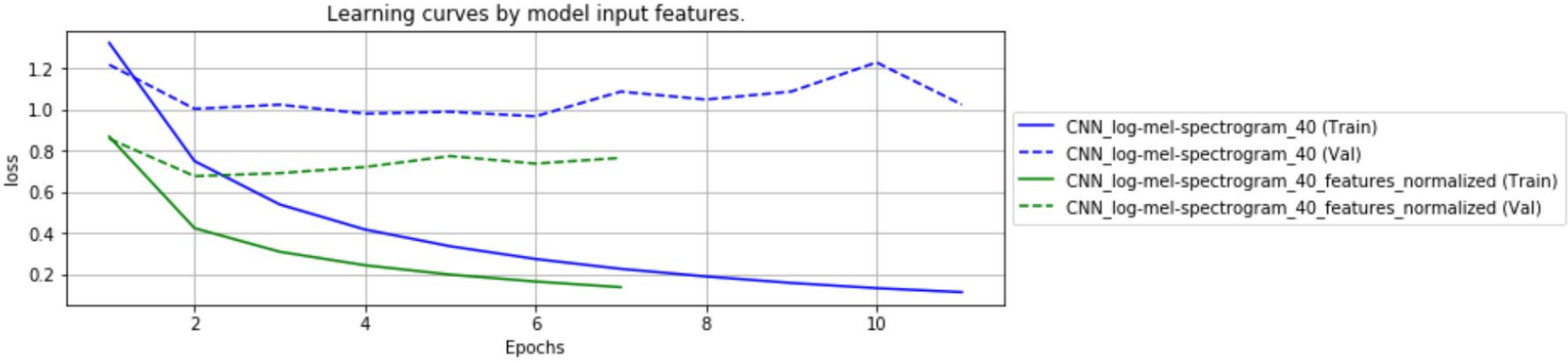
Early Stopping (in train & test for accuracy & loss)

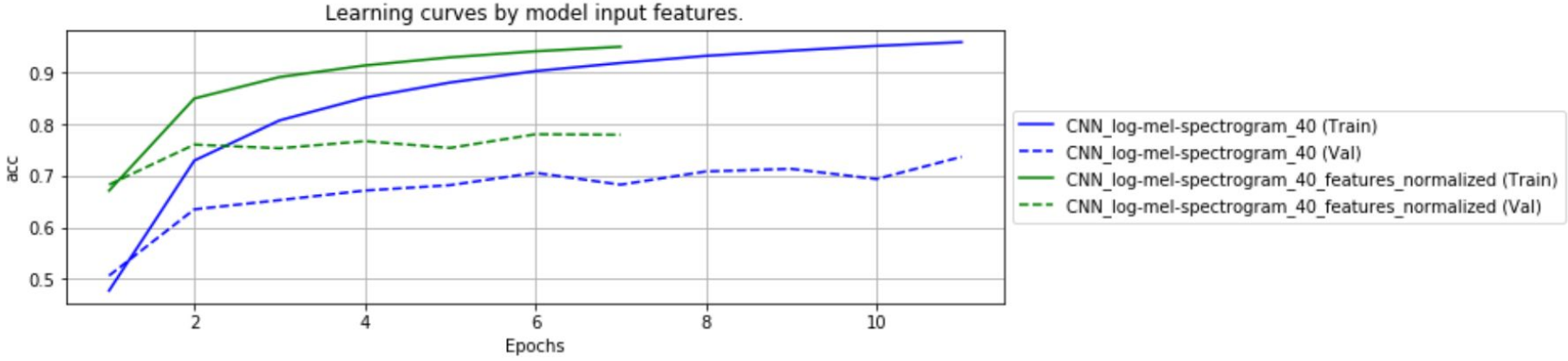
CNN - Results: without features normalization





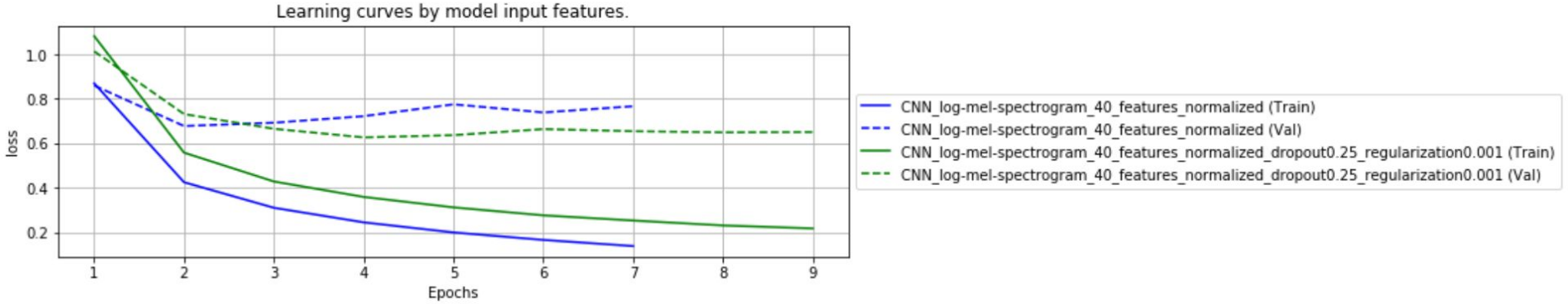
CNN - Results: with features normalization

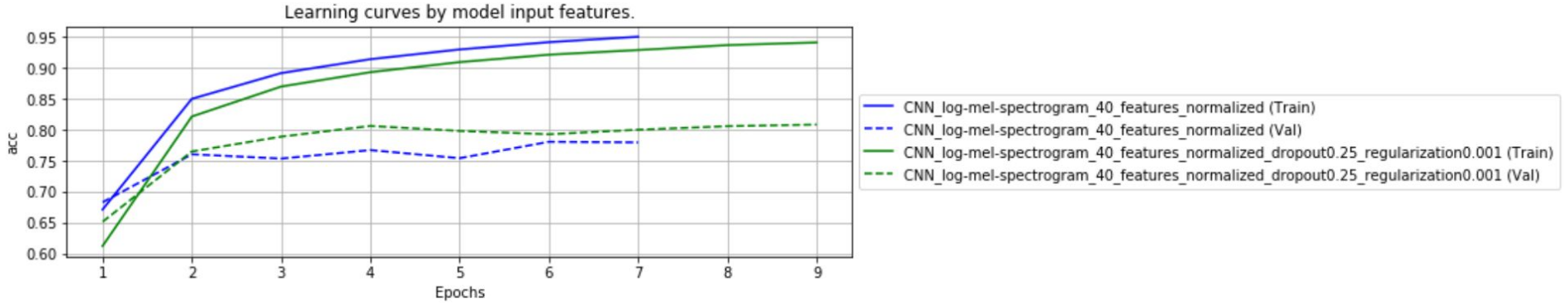




CNN - Results: regularization

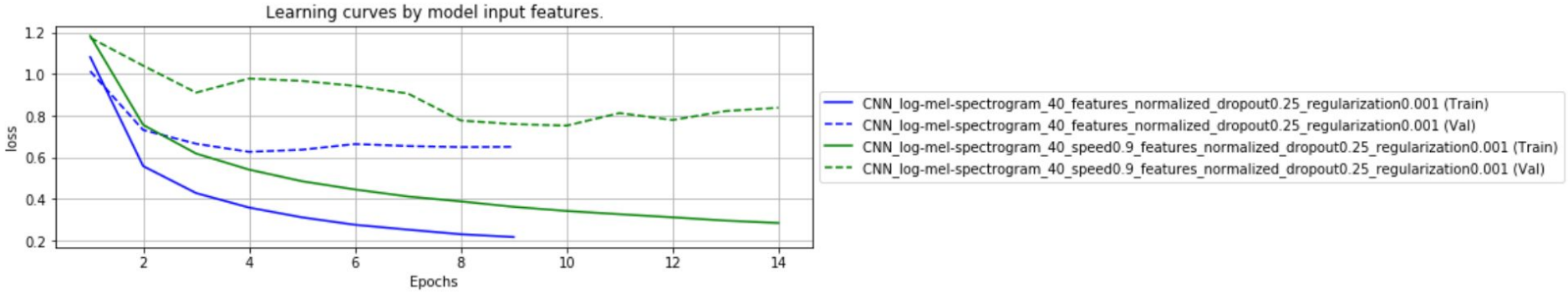
Dropout Layer 0.25 + Ridge Regularization 0.001

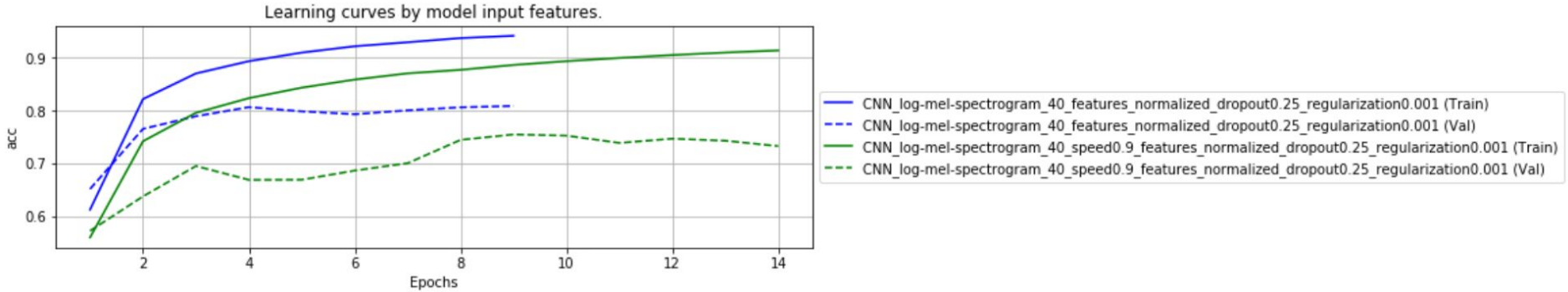




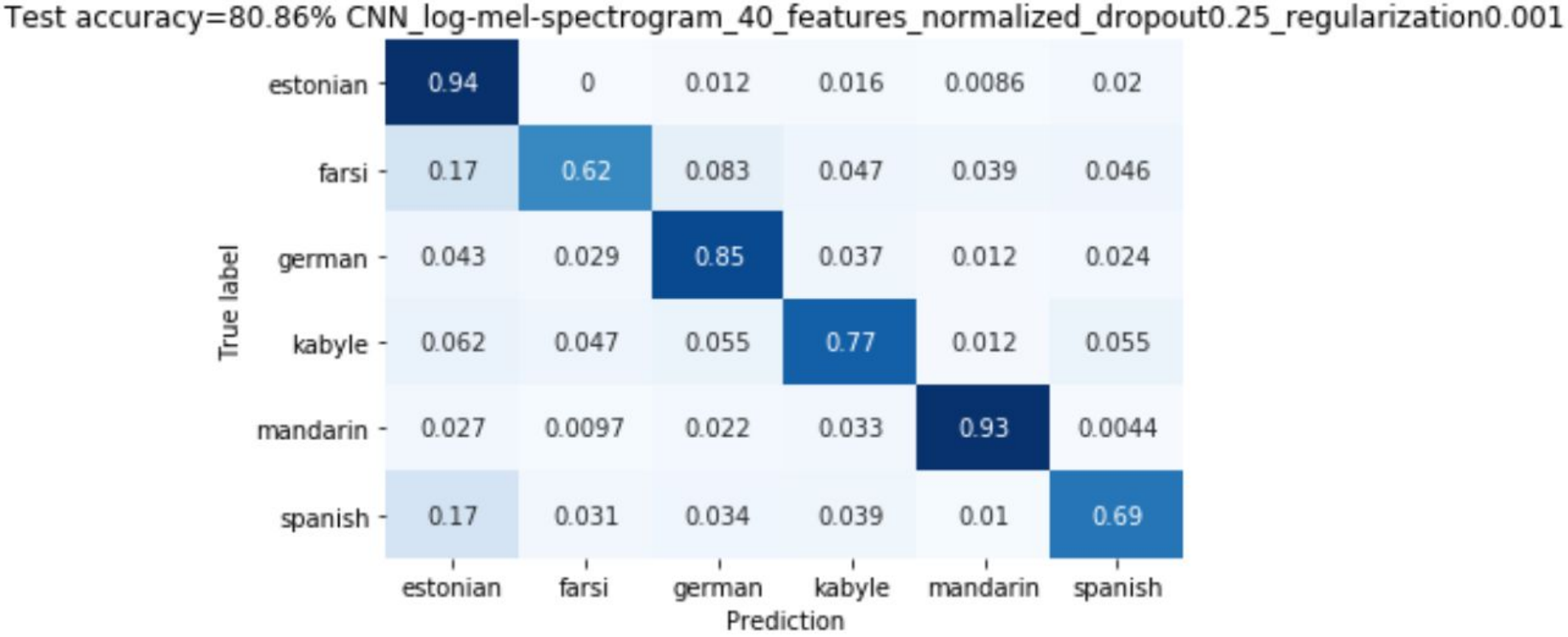
CNN - Results: speed down (FR=16kHz/0.9)

Speed=0.9 → Data augmentation X > 95k samples





Most difficult languages to distinct



Conclusions

80 N-Mels and MFCC does not improve results

Hop Splits moving window boosts the performance

Dropout + Ridge regularization reduce overfitting and leads to 80% acc

Speed down the audios does not improve our results

Thank you