Audio analysis: speech recognition

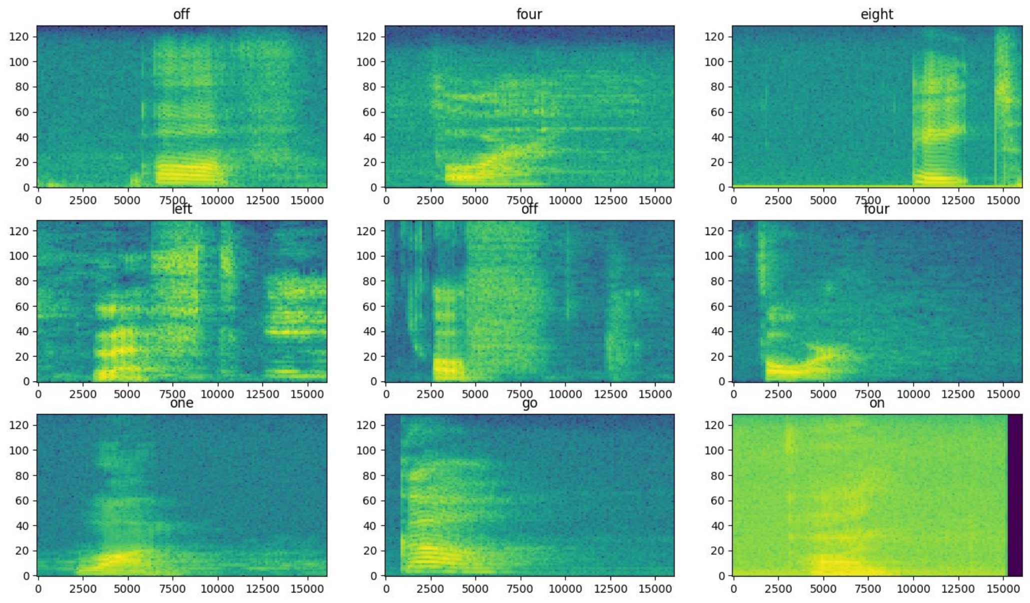
Carlota Sans and Mariola Monfort

Técnicas Avançadas de Análise de Dados
Supervisor: Filipe Veloso

1 2 9 0
UNIVERSIDADE D
COIMBRA

INDEX

- The data
- Our code
- Optimization
- Conclusions

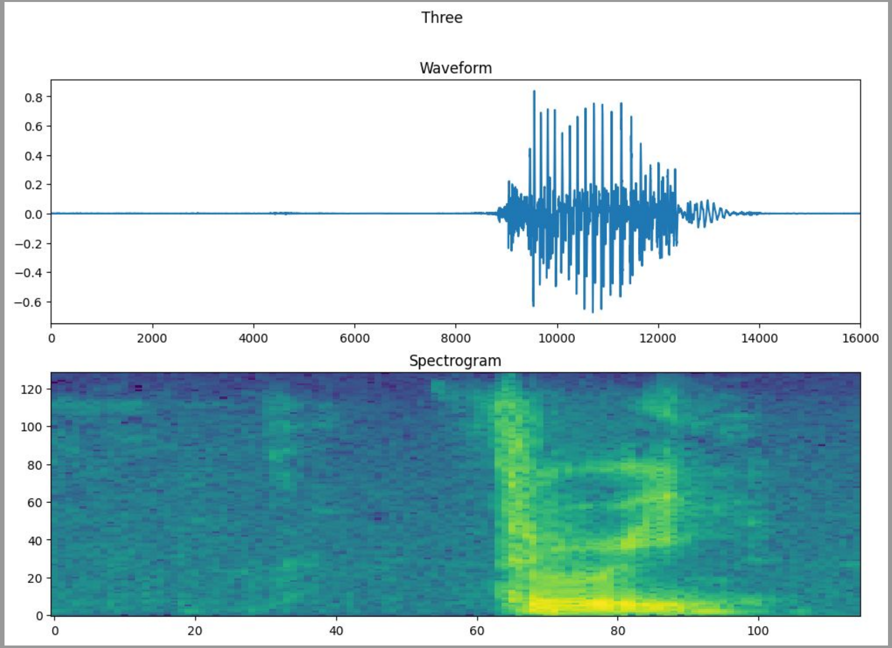


The Data

The goal of speech recognition is to transform human voice into text or instructions that a computer is able to process. It has many applications and it is very useful nowadays.

How do we analyze our data???

- Original data: .wav files
- Input data: spectrograms
- STFT



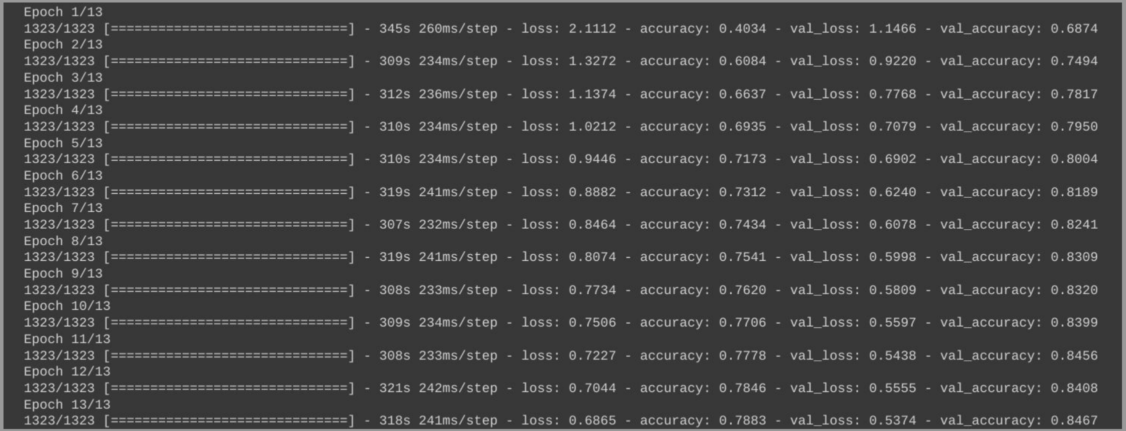
Our Code

How does the code work? How does it analyse the data?

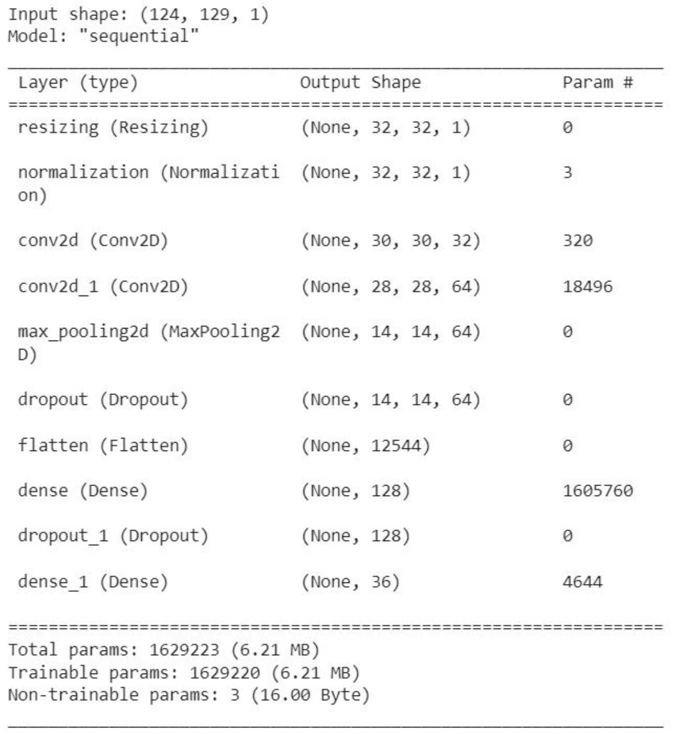
What were the initial parameters?

Our Code

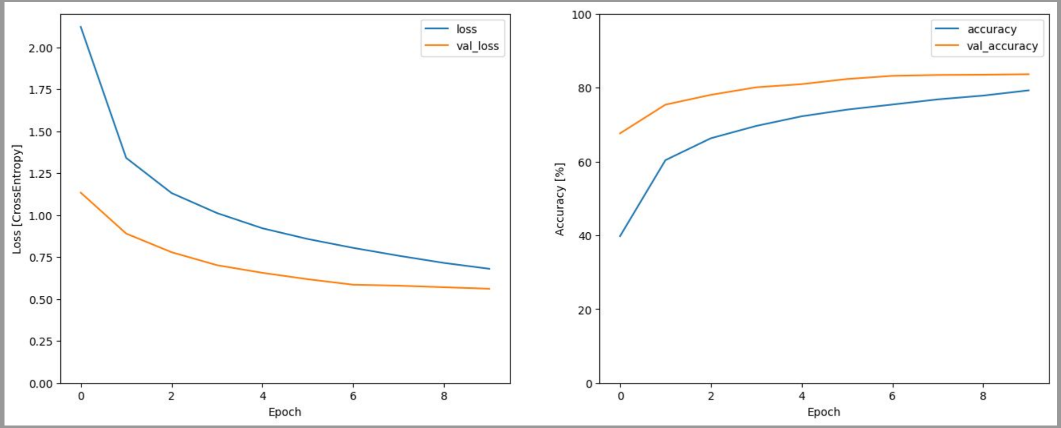
- Data split: 80% train, 10% validation, 10% test
- Transformation of the data
- Neural network: CNN
- Training of the model

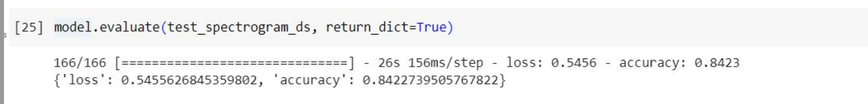


```
Epoch 1/13
1323/1323 [==============================] - 345s 260ms/step - loss: 2.1112 - accuracy: 0.4034 - val_loss: 1.1466 - val_accuracy: 0.6874
Epoch 2/13
1323/1323 [==============================] - 309s 234ms/step - loss: 1.3272 - accuracy: 0.6084 - val_loss: 0.9220 - val_accuracy: 0.7494
Epoch 3/13
1323/1323 [==============================] - 312s 236ms/step - loss: 1.1374 - accuracy: 0.6637 - val_loss: 0.7768 - val_accuracy: 0.7817
Epoch 4/13
1323/1323 [==============================] - 310s 234ms/step - loss: 1.0212 - accuracy: 0.6935 - val_loss: 0.7079 - val_accuracy: 0.7950
Epoch 5/13
1323/1323 [==============================] - 310s 234ms/step - loss: 0.9446 - accuracy: 0.7173 - val_loss: 0.6902 - val_accuracy: 0.8004
Epoch 6/13
1323/1323 [==============================] - 319s 241ms/step - loss: 0.8882 - accuracy: 0.7312 - val_loss: 0.6240 - val_accuracy: 0.8189
Epoch 7/13
1323/1323 [==============================] - 307s 232ms/step - loss: 0.8464 - accuracy: 0.7434 - val_loss: 0.6078 - val_accuracy: 0.8241
Epoch 8/13
1323/1323 [==============================] - 319s 241ms/step - loss: 0.8074 - accuracy: 0.7541 - val_loss: 0.5998 - val_accuracy: 0.8309
Epoch 9/13
1323/1323 [==============================] - 308s 233ms/step - loss: 0.7734 - accuracy: 0.7620 - val_loss: 0.5809 - val_accuracy: 0.8320
Epoch 10/13
1323/1323 [==============================] - 309s 234ms/step - loss: 0.7506 - accuracy: 0.7706 - val_loss: 0.5597 - val_accuracy: 0.8399
Epoch 11/13
1323/1323 [==============================] - 308s 233ms/step - loss: 0.7227 - accuracy: 0.7778 - val_loss: 0.5438 - val_accuracy: 0.8456
Epoch 12/13
1323/1323 [==============================] - 321s 242ms/step - loss: 0.7044 - accuracy: 0.7846 - val_loss: 0.5555 - val_accuracy: 0.8408
Epoch 13/13
1323/1323 [==============================] - 318s 241ms/step - loss: 0.6865 - accuracy: 0.7883 - val_loss: 0.5374 - val_accuracy: 0.8467
```

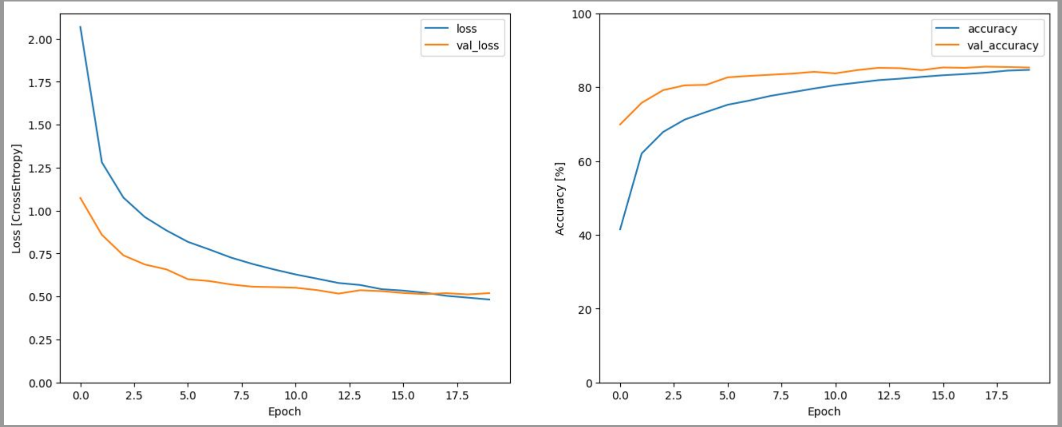


```
Input shape: (124, 129, 1)
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resizing (Resizing)         (None, 32, 32, 1)         0

 normalization (Normalizati  (None, 32, 32, 1)         3
 on)

 conv2d (Conv2D)             (None, 30, 30, 32)        320

 conv2d_1 (Conv2D)           (None, 28, 28, 64)        18496

 max_pooling2d (MaxPooling2  (None, 14, 14, 64)        0
 D)

 dropout (Dropout)           (None, 14, 14, 64)        0

 flatten (Flatten)           (None, 12544)             0

 dense (Dense)               (None, 128)               1605760

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 36)                4644

=================================================================
Total params: 1629223 (6.21 MB)
Trainable params: 1629220 (6.21 MB)
Non-trainable params: 3 (16.00 Byte)
_________________________________________________________________
```





```
[25] model.evaluate(test_spectrogram_ds, return_dict=True)

     166/166 [==============================] - 26s 156ms/step - loss: 0.5456 - accuracy: 0.8423
     {'loss': 0.5455626845359802, 'accuracy': 0.8422739505767822}
```



batch size = 64

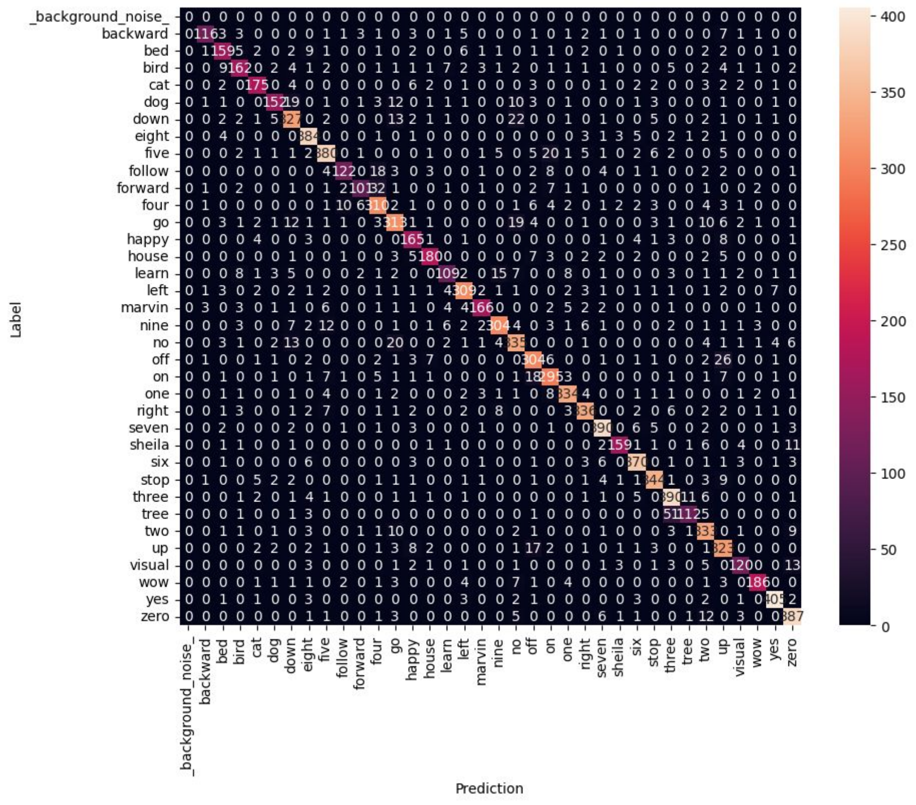
frame length= 225

frame step = 128

dropout rate 1 = 0.25

dropout rate 2 = 0.5

Our Code



	_background_noise_	backward	bed	bird	cat	dog	down	eight	five	follow	forward	four	go	happy	house	learn	left	marvin	nine	no	off	on	one	right	seven	sheila	six	stop	three	tree	two	up	visual	wow	yes	zero
_background_noise_	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
backward	0	116	3	3	0	0	0	0	1	1	3	1	0	3	0	1	5	0	0	0	1	0	1	2	1	0	1	0	0	0	0	7	1	1	0	0
bed	0	1	159	5	2	0	2	9	1	0	0	1	0	2	0	0	6	1	1	0	1	1	0	2	0	1	0	0	0	0	2	2	0	0	1	0
bird	0	0	9	162	0	2	4	1	2	0	0	1	1	1	1	7	2	3	1	2	0	1	1	1	1	0	0	0	5	0	2	4	1	1	0	2
cat	0	0	2	0	175	0	4	0	0	0	0	0	0	6	2	0	1	0	0	0	3	0	0	0	1	0	2	2	0	0	3	2	2	0	1	0
dog	0	1	1	0	0	152	19	0	1	0	1	3	12	0	1	1	1	0	0	10	3	0	1	0	0	0	1	3	0	0	0	1	0	0	1	0
down	0	0	2	2	1	5	327	0	2	0	0	0	13	2	1	1	0	0	0	22	0	0	1	0	1	0	0	5	0	0	2	1	0	1	1	0
eight	0	0	4	0	0	0	0	384	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	3	1	3	5	0	2	1	2	1	0	0	0	0
five	0	0	0	2	1	1	1	2	380	0	1	0	0	0	1	0	0	1	5	0	5	20	1	5	1	0	2	6	2	0	0	5	0	0	0	0
follow	0	0	0	0	0	0	0	0	4	122	0	18	3	0	3	0	0	1	0	0	2	8	0	0	4	0	1	1	0	0	2	2	0	0	0	1
forward	0	1	0	2	0	0	1	0	1	2	101	32	1	0	0	1	0	1	0	0	2	7	1	1	0	0	0	0	0	0	1	0	0	2	0	0
four	0	1	0	0	0	0	0	0	1	10	6	310	2	1	0	0	0	0	0	1	6	4	2	0	1	2	2	3	0	0	4	3	1	0	0	0
go	0	0	3	1	2	1	12	1	1	1	0	3	313	1	1	0	0	0	0	19	4	0	0	1	0	0	0	3	1	0	10	6	2	1	0	1
happy	0	0	0	0	4	0	0	3	0	0	0	0	0	165	1	0	1	0	0	0	0	0	0	0	1	0	4	1	3	0	0	8	0	0	0	1
house	0	0	0	0	0	0	1	0	0	1	0	0	3	5	180	0	0	0	0	0	7	3	0	2	2	0	2	0	0	0	2	5	0	0	0	0
learn	0	0	0	8	1	3	5	0	0	0	2	1	2	0	0	109	2	0	15	7	0	0	8	0	1	0	0	0	3	0	1	1	2	0	1	1
left	0	1	3	0	2	0	2	1	2	0	0	1	1	1	1	4	309	2	1	1	0	0	2	3	1	0	1	1	1	0	1	2	0	0	7	0
marvin	0	3	0	3	0	1	1	0	6	0	0	1	1	1	0	4	4	166	0	0	0	2	5	2	2	0	0	0	1	0	0	0	0	1	0	0
nine	0	0	0	3	0	0	7	2	12	0	0	0	1	0	1	6	2	2	304	4	0	3	1	6	1	0	0	0	2	0	1	1	1	3	0	0
no	0	0	3	1	0	2	13	0	0	0	0	0	20	0	0	2	1	1	4	335	0	0	0	1	0	0	0	0	0	0	4	1	1	1	4	6
off	0	1	0	0	1	1	0	2	0	0	0	2	1	3	7	0	0	0	0	0	304	6	0	0	1	0	1	1	0	0	2	26	0	0	1	0
on	0	0	1	0	0	1	0	1	7	1	0	5	1	1	1	0	0	0	0	1	18	295	3	0	0	0	0	0	1	0	1	7	0	0	1	0
one	0	0	1	1	0	0	1	0	4	0	0	1	2	0	0	0	2	3	1	1	0	8	334	4	0	0	1	1	1	0	0	0	1	2	0	1
right	0	0	1	3	0	0	1	2	7	0	0	1	1	2	0	0	2	0	8	0	0	0	3	336	0	0	2	0	6	0	2	2	0	1	1	0
seven	0	0	2	0	0	0	2	0	0	1	0	1	0	3	0	0	0	1	0	0	0	1	0	0	390	0	6	5	0	0	2	0	0	0	1	3
sheila	0	0	1	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	2	159	1	1	0	1	6	0	4	0	0	11
six	0	0	1	0	0	0	0	6	0	0	0	0	0	3	0	0	0	1	0	0	1	0	0	3	6	0	370	0	1	0	1	1	3	0	1	3
stop	0	1	0	0	5	2	2	0	0	0	0	1	1	1	0	0	1	1	0	0	1	0	0	1	4	1	1	344	1	0	3	9	0	0	0	0
three	0	0	0	1	2	0	1	4	1	0	0	0	1	1	1	0	1	0	0	0	0	0	0	1	1	0	5	0	390	11	6	0	0	0	0	1
tree	0	0	0	1	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	51	112	5	0	0	0	0	0
two	0	0	1	1	0	1	0	3	0	0	1	1	10	0	0	0	0	0	0	2	1	0	0	0	1	0	0	0	3	1	333	0	1	0	0	9
up	0	0	0	0	2	2	0	2	1	0	0	1	3	8	2	0	0	0	0	1	17	2	0	1	0	1	1	3	0	0	1	323	0	0	0	0
visual	0	0	0	0	0	0	0	3	0	0	0	0	1	2	1	0	1	0	0	1	0	0	0	0	1	3	0	1	3	0	5	0	120	0	0	13
wow	0	0	0	0	1	1	1	1	0	2	0	1	3	0	0	0	4	0	0	7	1	0	4	0	0	0	0	0	0	0	1	3	0	186	0	0
yes	0	0	1	0	1	0	0	3	0	0	0	0	0	0	0	0	3	0	0	2	1	0	0	0	0	0	3	0	0	0	2	0	1	0	405	2
zero	0	0	0	0	0	0	0	1	1	0	0	1	3	0	0	0	0	0	0	5	0	0	0	0	6	1	1	0	0	1	12	0	3	0	0	387

batch size = 64

frame length = 225

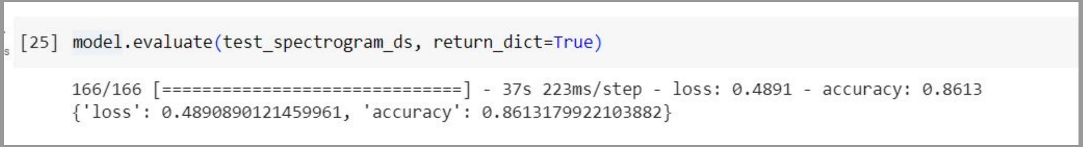
frame step = 128

dropout rate 1 = 0.25

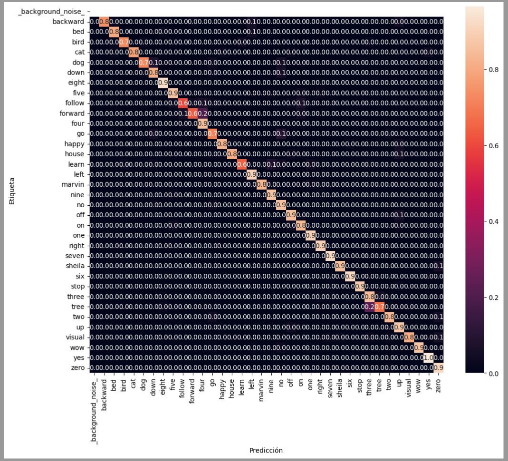
dropout rate 2 = 0.5

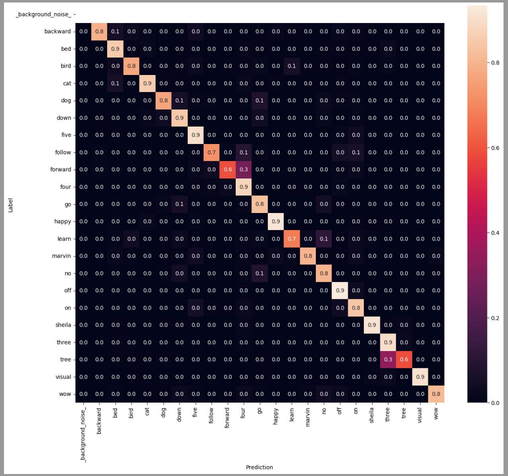
Optimization

1. Set the number of epochs to 13
2. Understand the confusion matrix
3. Reduce the data
4. Use Optuna to find the best parameters
   - dropout rate = 0.4315 and 0,4661
   - batch size = 64
   - frame step = 128
5. Apply them to the whole data set



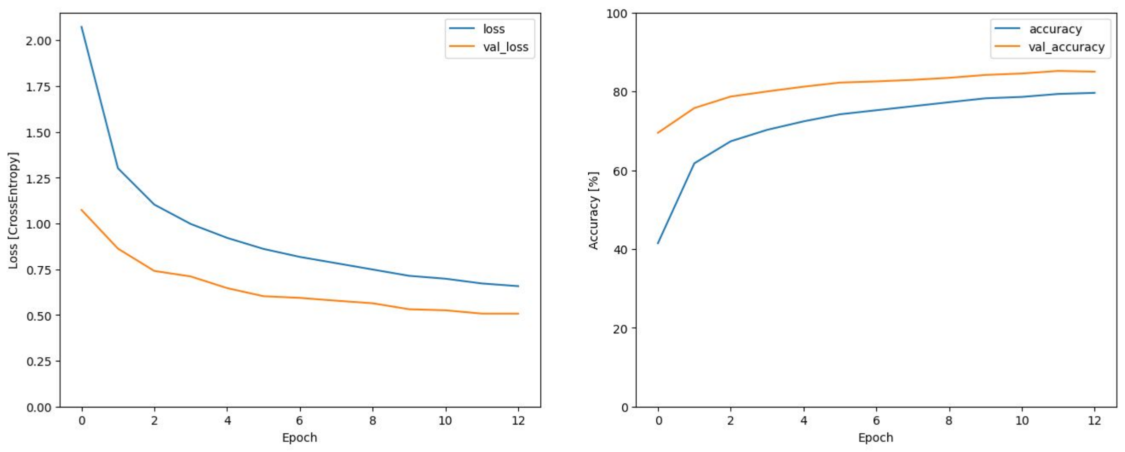
```
[25] model.evaluate(test_spectrogram_ds, return_dict=True)

     166/166 [==============================] - 37s 223ms/step - loss: 0.4891 - accuracy: 0.8613
     {'loss': 0.4890890121459961, 'accuracy': 0.8613179922103882}
```

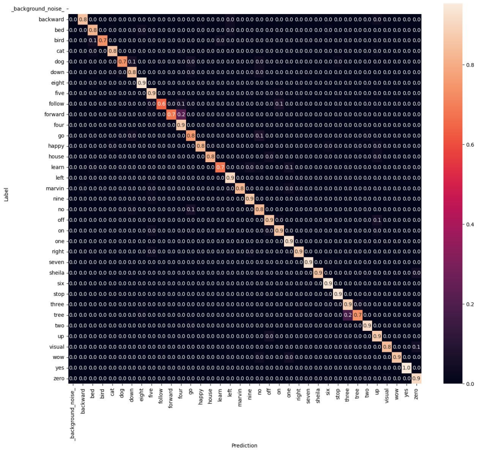




Conclusions

- Improvement of the loss (0.06) and accuracy (2%).
- Understanding of how machine learning works and how the parameters within our code affect the final result.


loss
val_loss
Loss [CrossEntropy]
Epoch
accuracy
val_accuracy
Accuracy [%]


Label
_background_noise_
backward
bed
bird
cat
dog
down
eight
five
follow
forward
four
go
happy
house
learn
left
marvin
nine
no
off
on
one
right
seven
sheila
six
stop
three
tree
two
up
visual
wow
yes
zero
Prediction