Speech to Text Digit Recognition

Mollon, Manuel 58023
Vijande, Ezequiel 58057

Introducción y Objetivo

Introducción al problema

- Se quiere crear un algoritmo que pueda traducir dígitos hablados en un audio a número.
- Este mismo puede servir para traducción entre idiomas o como herramienta de acceso en un sistema de seguridad.
- Se utilizará una red neuronal (CNN) para predecir los dígitos a partir de un audio.
- La red tendrá como inputs espectrogramas en escala mel, transformando el problema en uno de reconocimiento de imágenes.

Dataset & Pre Processing

Procedimiento

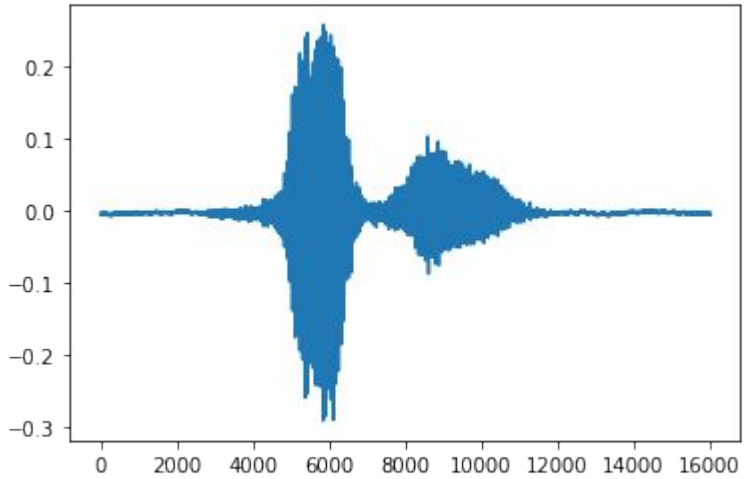
- Abrir dataset con wavs
- Calcular espectrograma mel
- Pasar espectrograma a jpg
- Guardar jpg en la nube en carpetas con label

Tensorflow Data Speech Commands

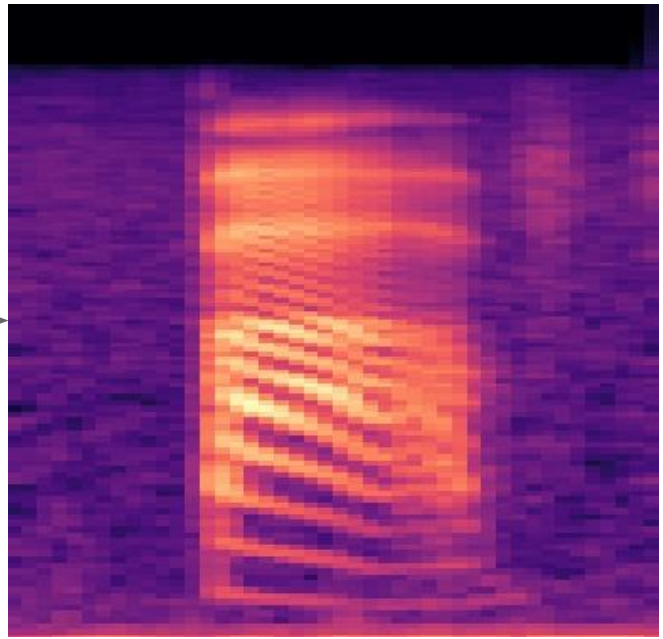
Dataset Distribution:

zero : 2376
one : 2370
two : 2373
three : 2356
four : 2372
five : 2357
six : 2369
seven : 2377
eight : 2352
nine : 2364

WAV



MEL SPECTROGRAM



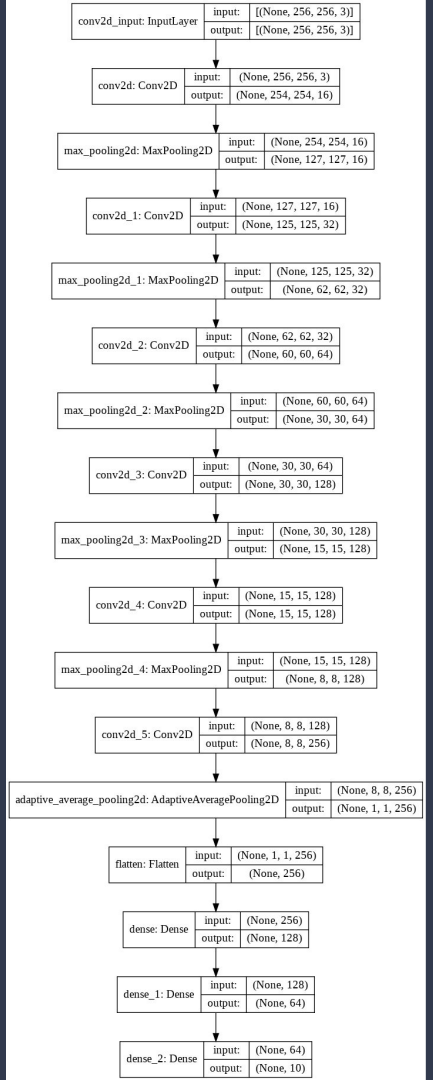
http://download.tensorflow.org/data/speech_commands_v0.01.tar.gz

Model definition

Model: "sequential"

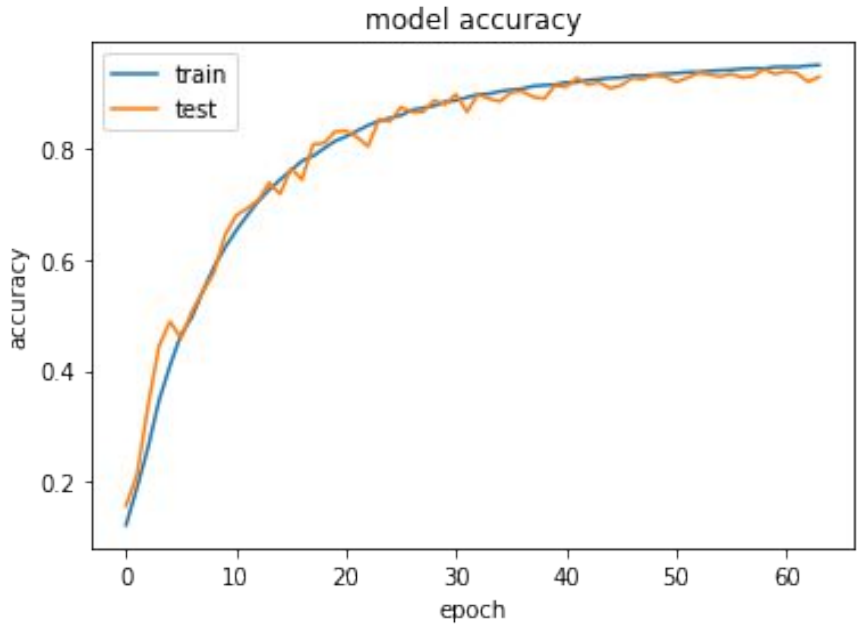
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 16)	448
max_pooling2d (MaxPooling2D)	(None, 127, 127, 16)	0
conv2d_1 (Conv2D)	(None, 125, 125, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 30, 30, 64)	0
conv2d_3 (Conv2D)	(None, 30, 30, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 15, 15, 128)	0
conv2d_4 (Conv2D)	(None, 15, 15, 128)	147584
max_pooling2d_4 (MaxPooling2	(None, 8, 8, 128)	0
conv2d_5 (Conv2D)	(None, 8, 8, 256)	295168
adaptive_average_pooling2d (	(None, 1, 1, 256)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 10)	650

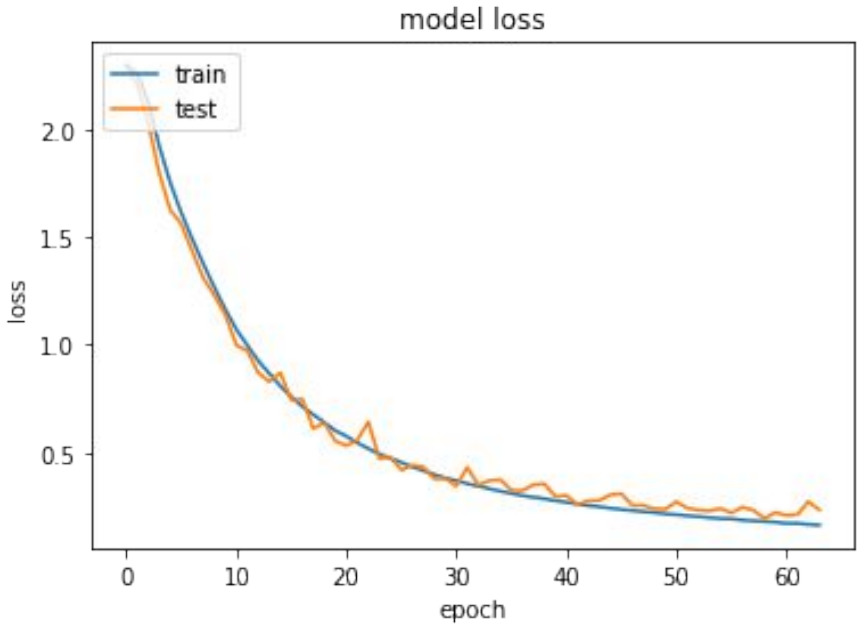
Total params: 581,994
Trainable params: 581,994
Non-trainable params: 0



Training & Validation

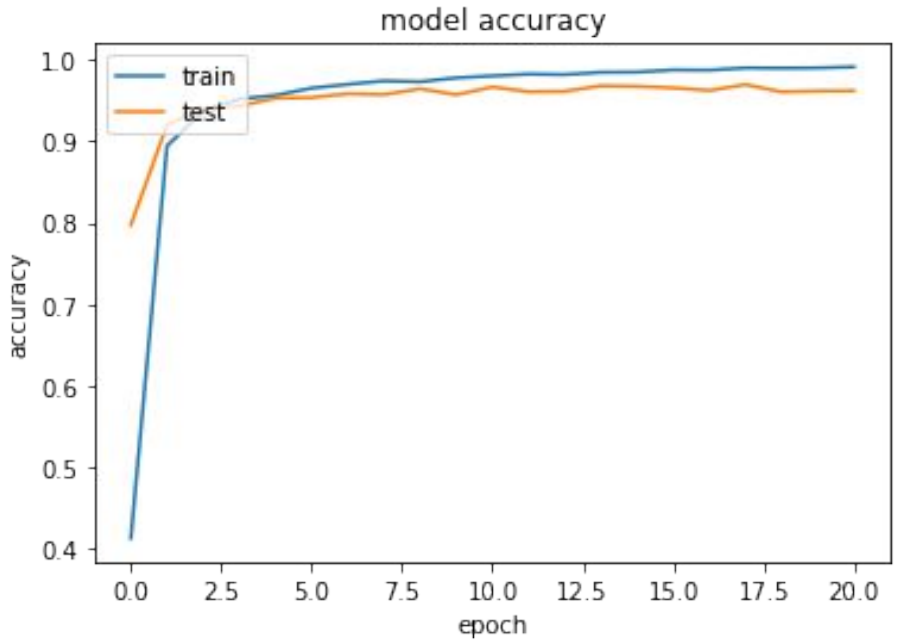
Entrenamiento con RMSprop Optimizer

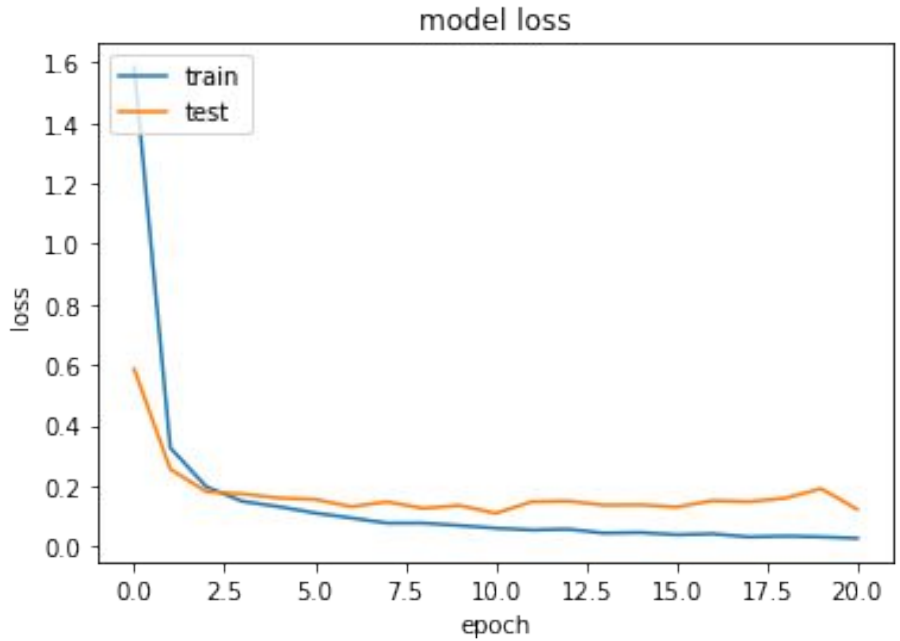




93% en testing

Entrenamiento con Adam Optimizer



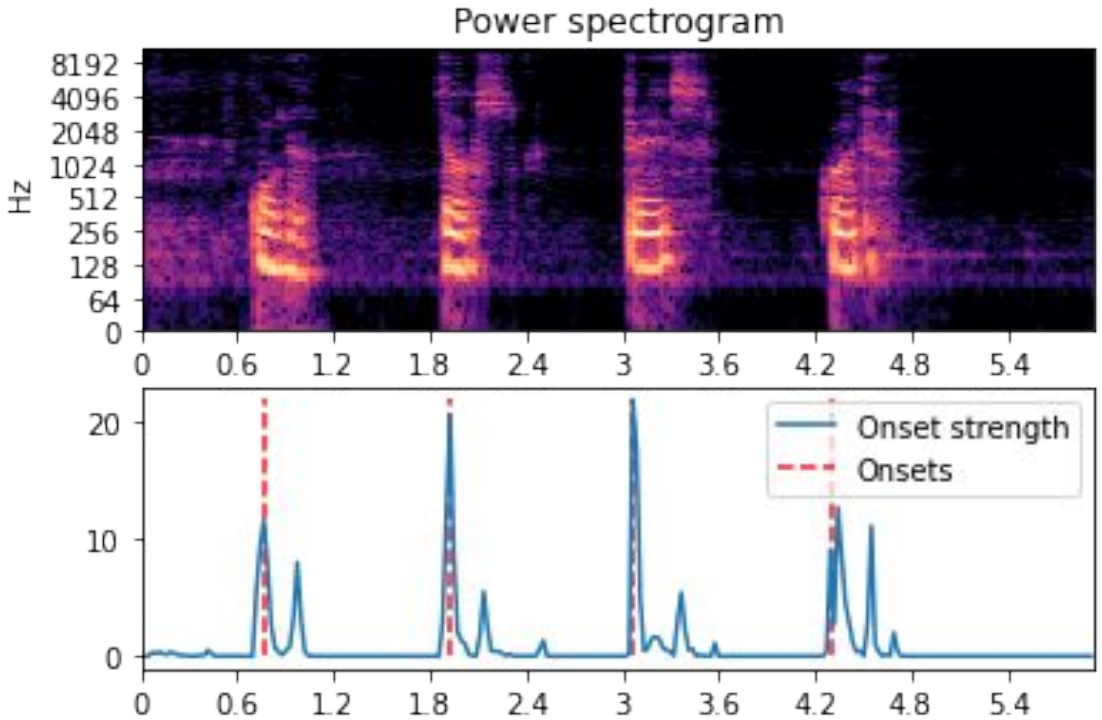


96% en testing

Real Time Processing

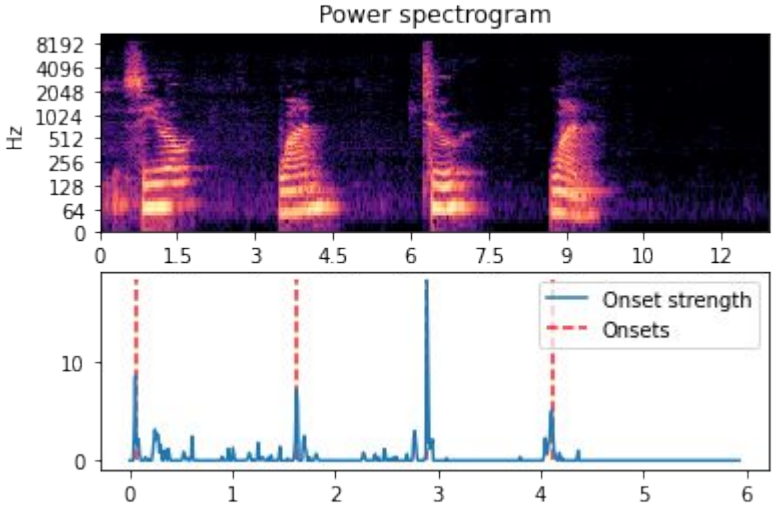
Onset detection

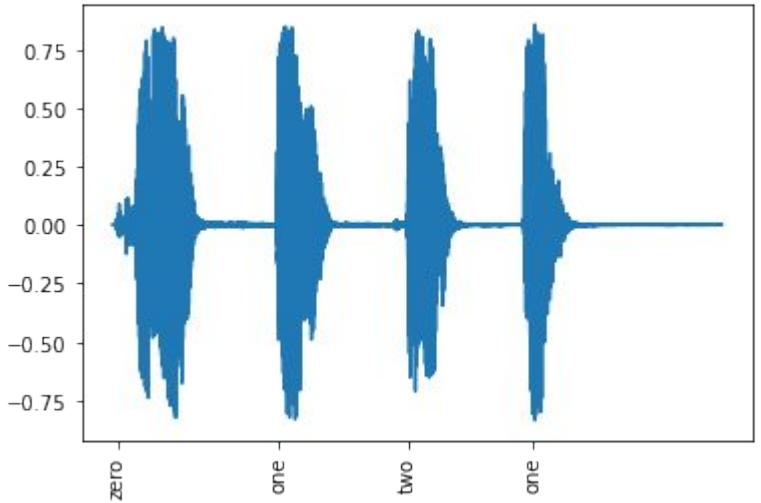
- Se graba un audio de 7 segundos
- Se calcula el onset strength de la señal original
- Se utiliza la función peak detect sobre la señal calculada anteriormente.
- Se toman 0.3*sr antes y 0.7*sr después de los picos detectados para obtener un segundo de audio
- Se corta el audio en audios pequeños de un segundo
- Se pasan los audios obtenidos a jpg como se hizo anteriormente y se infiere con el modelo entrenado los dígitos hablados.



Testing

Ejemplo


Power spectrogram
Hz
8192
4096
2048
1024
512
256
128
64
0
0
1.5
3
4.5
6
7.5
9
10
12
Onset strength
Onsets
10
0
0
1
2
3
4
5
6


0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
zero
one
two
one