1) SPECTRE: SPECTROGRAM BASED MUSIC RECOMMENDER

- **WHAT**
  A SONG RECOMMENDER TO LET ME LISTEN TO MY MUSIC ON THE STEREO, BUT ONLY PLAYING SONGS THAT DO NOT UPSET MY FAMILY

- **WHY**
  EXISTING RECOMMENDER SYSTEMS ARE BASED ON META DATA AND PROFILE MATCHING BETWEEN PERSONS. THIS IS TOO COARSE-GRAINED FOR MY PURPOSES. I WANT TO BUILD A SYSTEM TO RECOMMEND SONGS BASED ON THEIR SOUND CHARACTERISTICS

- **HOW**
  TRANSFORM MP3 FILES INTO SPECTROGRAMS TO TRAIN A CONVOLUTIONAL NEURAL NETWORK

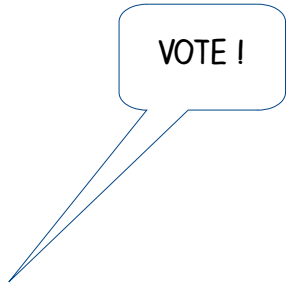
2) LABELLING THE DATA

MY WIFE NEEDS TO LABEL MY TRAINING DATA. FOR THAT PURPOSE, A PROGRAM PLAYS RANDOM SONGS FROM POSITION 60S FOR MAX 30S. SHE CAN PRESS ‹Y› OR ‹N› ANY TIME TO CONTINUE TO THE NEXT SONG.
SHE WAS ABLE TO LABEL 1000 SONGS WITHIN 40 MINUTES!

MP3 COLLECTION, ~ 10K SONGS

PLAY RANDOM SONGS

VOTE !



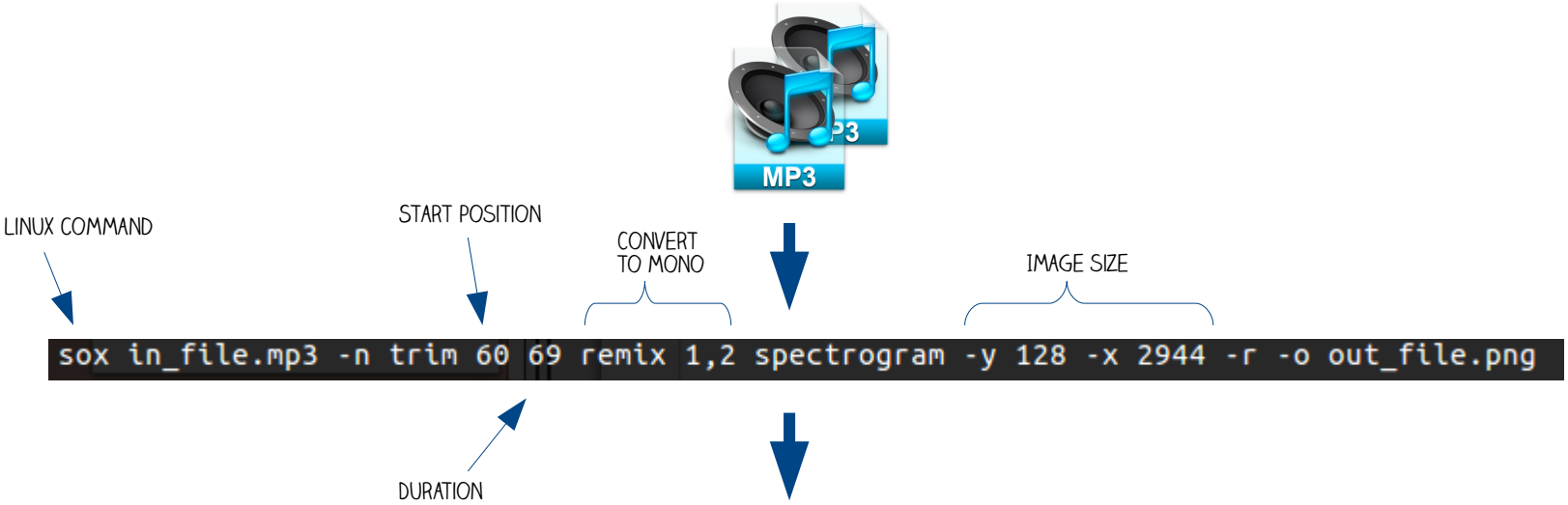
	A	D
1	/mp3/Prince/Prince - Rock This Joint - CD 2/05 - Release Yourself (incl. Hypno Paradise).mp3	n
2	/mp3/The Stooges/The Stooges - The Weirdness/05 - The Weirdness.mp3	n
3	/mp3/Prince/Princes iVault/1976/Sweet Thing.mp3	n
4	/mp3/Wallis Bird/Wallis Bird - The Mistakes Are Intentional/06 - Who's Listening Now.mp3	y
5	/mp3/Stiller Has/Stiller Has - Geisterbahn/10 - Sunnebrülle.mp3	y
6	/mp3/Madonna/Madonna - Ray Of Light/07 - Madonna - Sky Fits Heaven.mp3	n
7	/mp3/Boy George/Boy George - Sold/10 - We've Got The Right.mp3	n
8	/mp3/Iggy Pop/Iggy Pop - A Million In Prizes - The Anthology - CD1/13 - Funtime.mp3	n
9	/mp3/The Brand New Heavies/The Brand New Heavies - Trunk Funk The Best Of The Brand New Heavies/05 - Dream On Dreamer.mp3	y
10	/mp3/Gorillaz/Gorillaz - Gorillaz/13 - Starshine.mp3	y
11	/mp3/Prince/The Slaughterhouse/06-Props_N_Pounds.mp3	y
12	/mp3/Prince/Princes iVault/1982/Something In The Water (Does Not Com 1.mp3	n
13	/mp3/Iggy Pop/Iggy Pop - A Million In Prizes - The Anthology - CD1/11 - Kill City.mp3	n
14	/mp3/SOS - SOS/06 - SOS - Don't say Shit (P Chrétien).mp3	n
15	/mp3/Various - Putumayo Presents Paris/05 - Je reste au lit.mp3	y

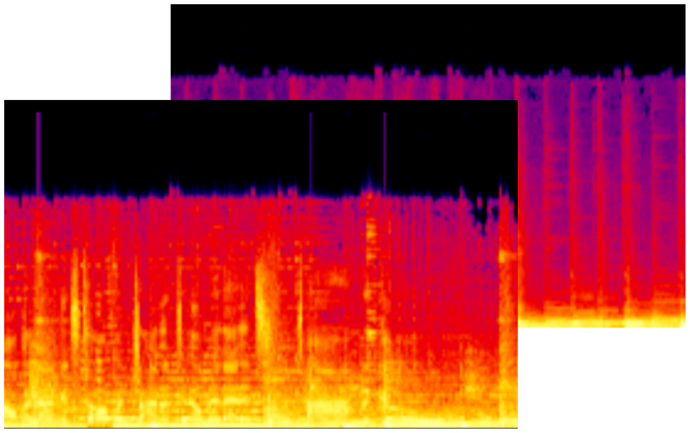
~1000 SAMPLES FOR SUPERVISED LEARNING

~500 SONGS

~500 SONGS

3) DATA PREPARATION



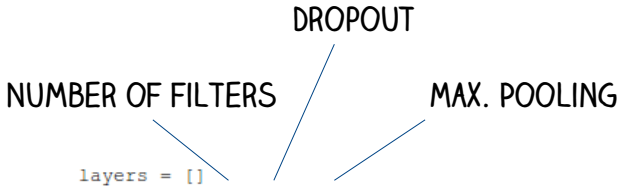


TAKE A 69 SECONDS LONG PIECE FROM EACH SONG AND CONVERT TO A SPECTROGRAM IMAGE

4) AUGMENTATION

SPLIT EACH IMAGE FROM 3) INTO MULTIPLE SAMPLES TO HAVE MORE TRAINING DATA AVAILABLE

5) AUTOMATED EVALUATION OF NETWORKS




```
layers = []
layers += [[16],[0],[0]]
layers += [[0,16],[0],[1]]
layers += [[0,16],[0],[0]]
layers += [[0,16,32],[0],[1]]
layers += [[0,16,32],[0],[0]]
layers += [[0,16,32],[0],[1]]
layers += [[0,16,64],[0],[0]]
layers += [[0,16,64],[0],[1]]
layers += [[0,16,64],[0],[0]]
layers += [[0,16,64],[0],[1]]
layers += [[400],[0.5],[0]]
```

- DEFINITION OF 10 CONVOLUTIONAL LAYERS IN SCRIPT
- EACH LAYER WITH VARYING NUMBER OF FILTERS
- BRUTE FORCE SEARCH FOR BEST CONFIGURATION
- SELECTION BY MAX ACCURACY (RESULTS WRITTEN TO LOGFILE)

```
def add_c_layer(model, filter_param, dropout_param,max_pool_param):
    if filter_param > 0:
        model.add(Convolution2D(filter_param, kernel_size, input_shape=my_input_shape, padding='same'))
        model.add(BatchNormalization())
        model.add(Activation('relu'))
        if max_pool_param > 0:
            model.add(MaxPooling2D(pool_size=pool_size))
        if dropout_param > 0:
            model.add(Dropout(dropout_param))
```

EarlyStopping [source]

```
keras.callbacks.EarlyStopping(monitor='val_loss', min_delta=0, patience=0, verbose=0, mode='auto')
```

Stop training when a monitored quantity has stopped improving.

Arguments

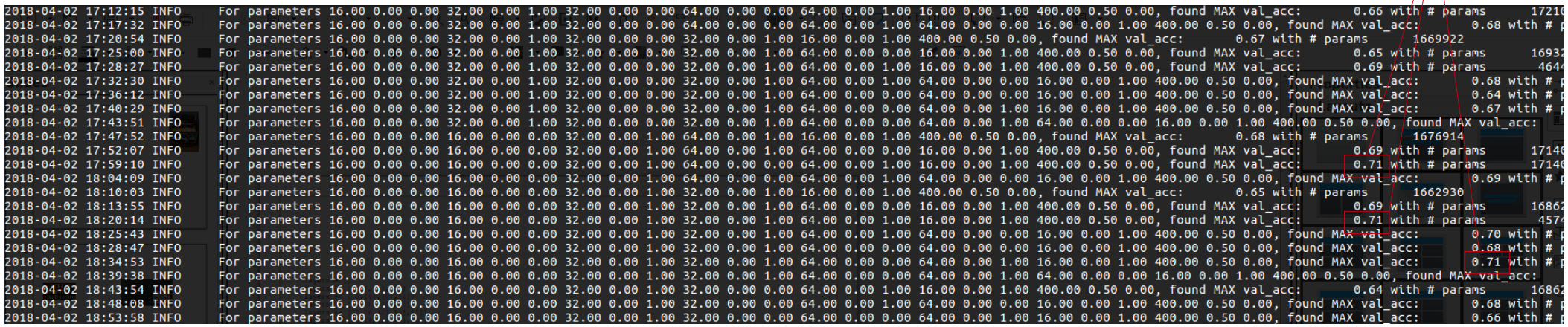
- **monitor**: quantity to be monitored.
- **min_delta**: minimum change in the monitored quantity to qualify as an improvement, i.e. an absolute change of less than min_delta, will count as no improvement.
- **patience**: number of epochs with no improvement after which training will be stopped.
- **verbose**: verbosity mode.
- **mode**: one of {auto, min, max}. In `min` mode, training will stop when the quantity monitored has stopped decreasing; in `max` mode it will stop when the quantity monitored has stopped increasing; in `auto` mode, the direction is automatically inferred from the name of the monitored quantity.

STOP EVALUATION WHEN NO PROGRESS FOR THREE EPOCS

6) NETWORK SELECTION

STEP 1: FIND NETWORK WITH MAX ACCURACY FROM LOGGED RESULTS

MAX ACCURACY



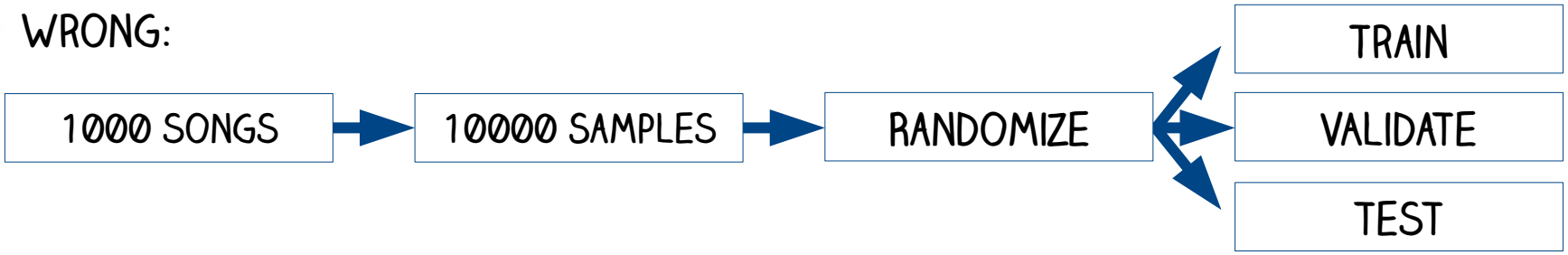
STEP 2: TRIAL-AND-ERROR FINE TUNING OF THOSE NETWORKS

7) LESSONS LEARNED: BE CAREFUL WITH DATA SEPARATION!

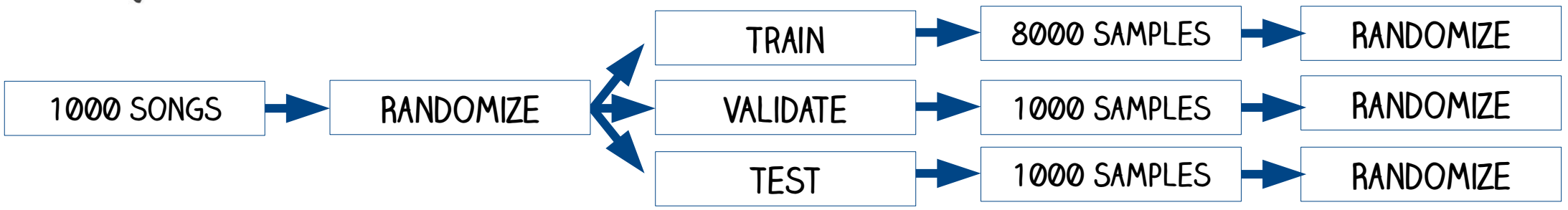
→ ACCURACY OF 73% IN THE FIRST TRIES. THIS VALUE IS SUSPICIALLY HIGH

→ I HAD THE SAME (WELL, VERY SIMILAR) DATA IN THE TRAINING AND THE VALIDATION SETS

WRONG:

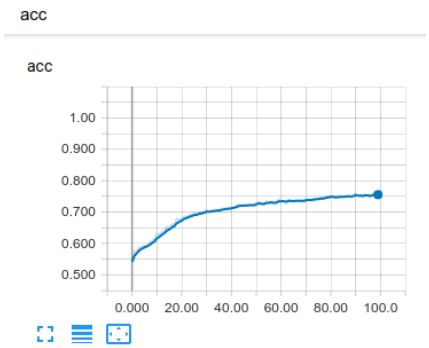


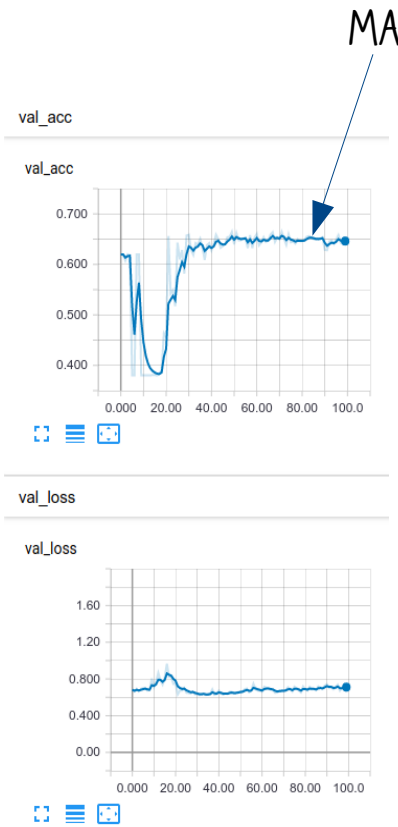
CORRECT:



AND TRY AGAIN...

8) EVALUATION ON TEST DATA






```
Layer (type)                 Output Shape              Param #
conv2d_1 (Conv2D)            (None, 128, 128, 16)      448
batch_normalization_1 (Batch (None, 128, 128, 16)      64
activation_1 (Activation)    (None, 128, 128, 16)      0
dropout_1 (Dropout)          (None, 128, 128, 16)      0
conv2d_2 (Conv2D)            (None, 128, 128, 16)      2320
batch_normalization_2 (Batch (None, 128, 128, 16)      64
activation_2 (Activation)    (None, 128, 128, 16)      0
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 16)        0
dropout_2 (Dropout)          (None, 64, 64, 16)        0
conv2d_3 (Conv2D)            (None, 64, 64, 16)        2320
batch_normalization_3 (Batch (None, 64, 64, 16)        64
activation_3 (Activation)    (None, 64, 64, 16)        0
dropout_3 (Dropout)          (None, 64, 64, 16)        0
conv2d_4 (Conv2D)            (None, 64, 64, 16)        2320
batch_normalization_4 (Batch (None, 64, 64, 16)        64
activation_4 (Activation)    (None, 64, 64, 16)        0
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 16)        0
dropout_4 (Dropout)          (None, 32, 32, 16)        0
conv2d_5 (Conv2D)            (None, 32, 32, 16)        2320
batch_normalization_5 (Batch (None, 32, 32, 16)        64
activation_5 (Activation)    (None, 32, 32, 16)        0
dropout_5 (Dropout)          (None, 32, 32, 16)        0
conv2d_6 (Conv2D)            (None, 32, 32, 16)        2320
batch_normalization_6 (Batch (None, 32, 32, 16)        64
activation_6 (Activation)    (None, 32, 32, 16)        0
max_pooling2d_3 (MaxPooling2 (None, 16, 16, 16)        0
dropout_6 (Dropout)          (None, 16, 16, 16)        0
conv2d_7 (Conv2D)            (None, 16, 16, 16)        2320
batch_normalization_7 (Batch (None, 16, 16, 16)        64
activation_7 (Activation)    (None, 16, 16, 16)        0
dropout_7 (Dropout)          (None, 16, 16, 16)        0
conv2d_8 (Conv2D)            (None, 16, 16, 16)        2320
batch_normalization_8 (Batch (None, 16, 16, 16)        64
activation_8 (Activation)    (None, 16, 16, 16)        0
max_pooling2d_4 (MaxPooling2 (None, 8, 8, 16)          0
dropout_8 (Dropout)          (None, 8, 8, 16)          0
flatten_1 (Flatten)          (None, 1024)              0
dense_1 (Dense)              (None, 400)               410000
batch_normalization_9 (Batch (None, 400)               1600
activation_9 (Activation)    (None, 400)               0
dropout_9 (Dropout)          (None, 400)               0
dense_2 (Dense)              (None, 2)                 802
activation_10 (Activation)   (None, 2)                 0
=================================================================
Total params: 429,602
Trainable params: 428,546
Non-trainable params: 1,056
```

FINAL EVALUATION ON TEST DATA SET: 1230 SAMPLES, ACCURACY=0.69!

NEEDS STILL IMPROVEMENT FOR PRODUCTIVE USE ;-)