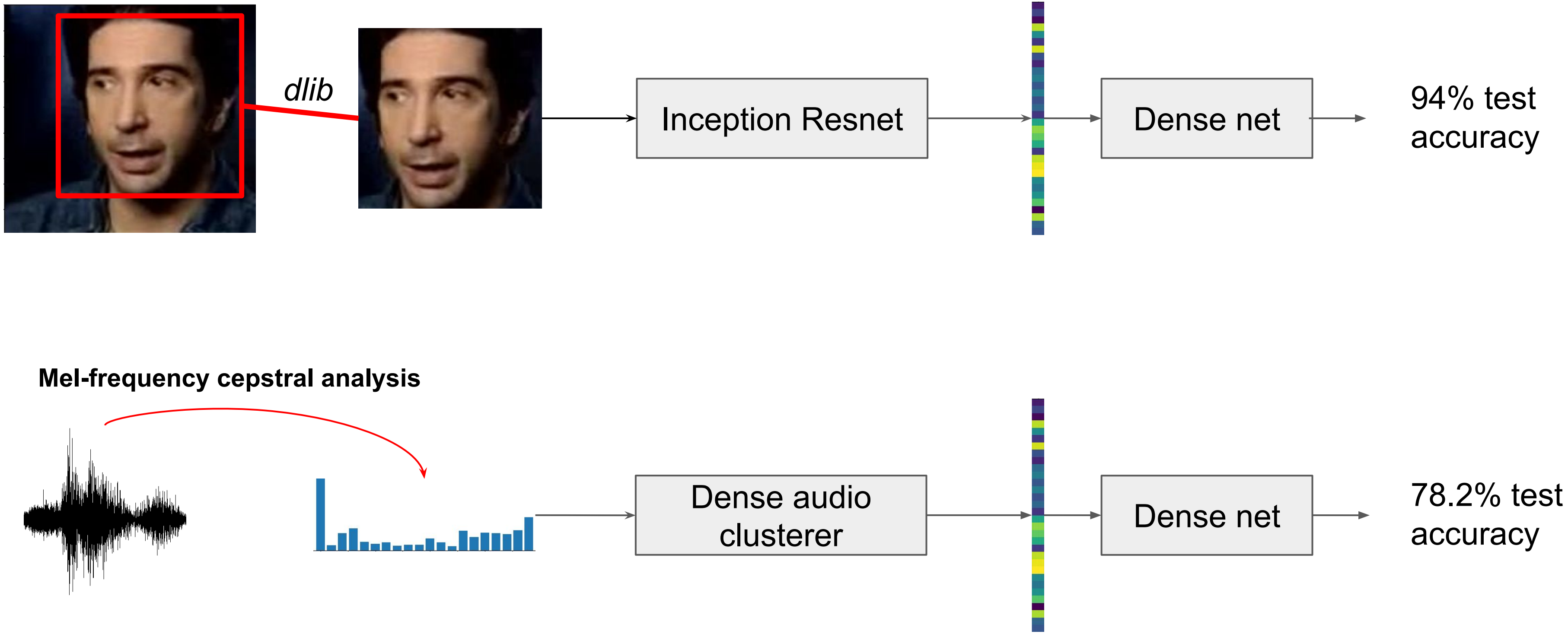
# Group 5

Frédéric Dux
Louis Suter
Utsav Akhaury

# General pipelines for images and audio



dlib

Inception Resnet

Dense net

94% test accuracy

Mel-frequency cepstral analysis

Dense audio clusterer

Dense net

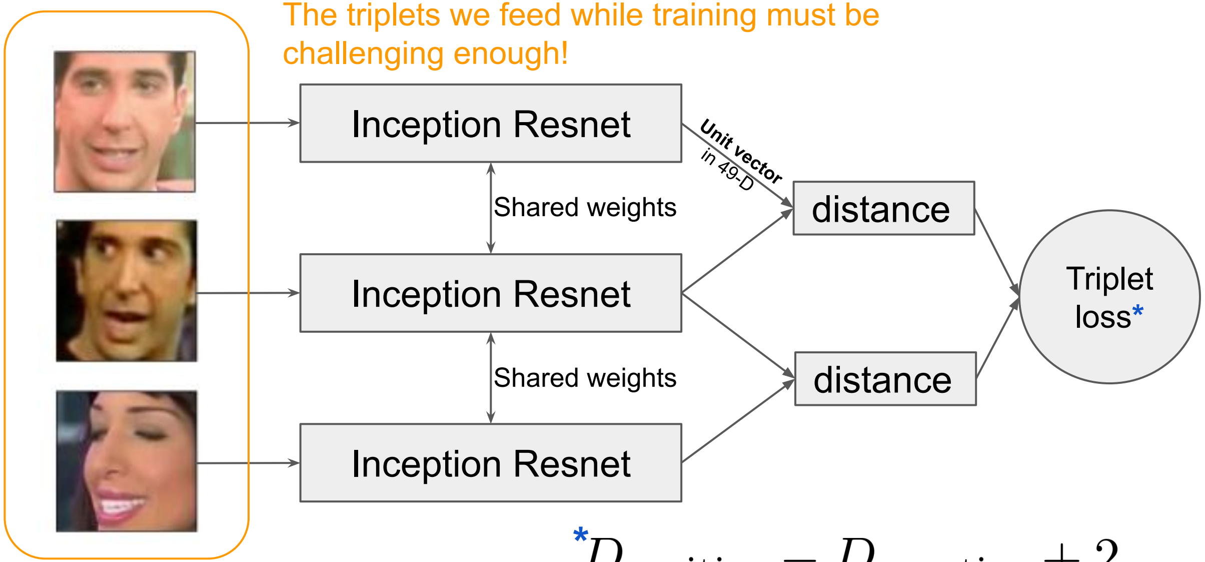
78.2% test accuracy

# Image Part: Training the Encoder in two steps

1 - Regular cross-entropy loss training → Gets us to roughly 80% test accuracy

2 - Triplet loss training

The triplets we feed while training must be challenging enough!



$$^{*}D_{\text{positive}} - D_{\text{negative}} + 2$$

Resulting encodings

Dense net

Gets us to roughly 94% test accuracy

# Audio Part: Feeding & Training the Encoder

Calculate the Cepstral coefficients (MFCC, chroma, mel, contrast)

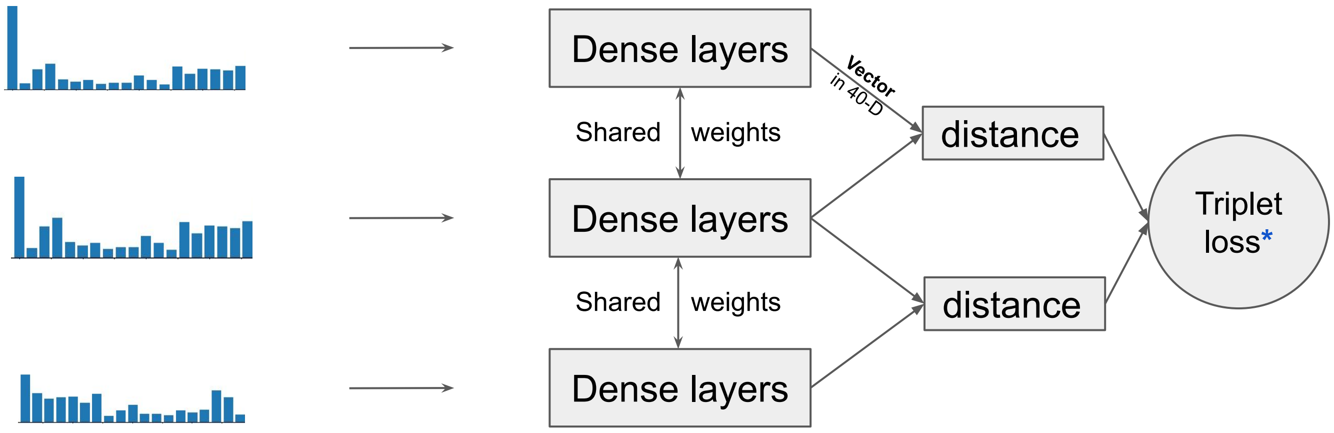
using
*librosa*

Triplet loss training

To enhance the clustering of the audio features

Tune the Dropouts

**We average the coefficients over the time windows**



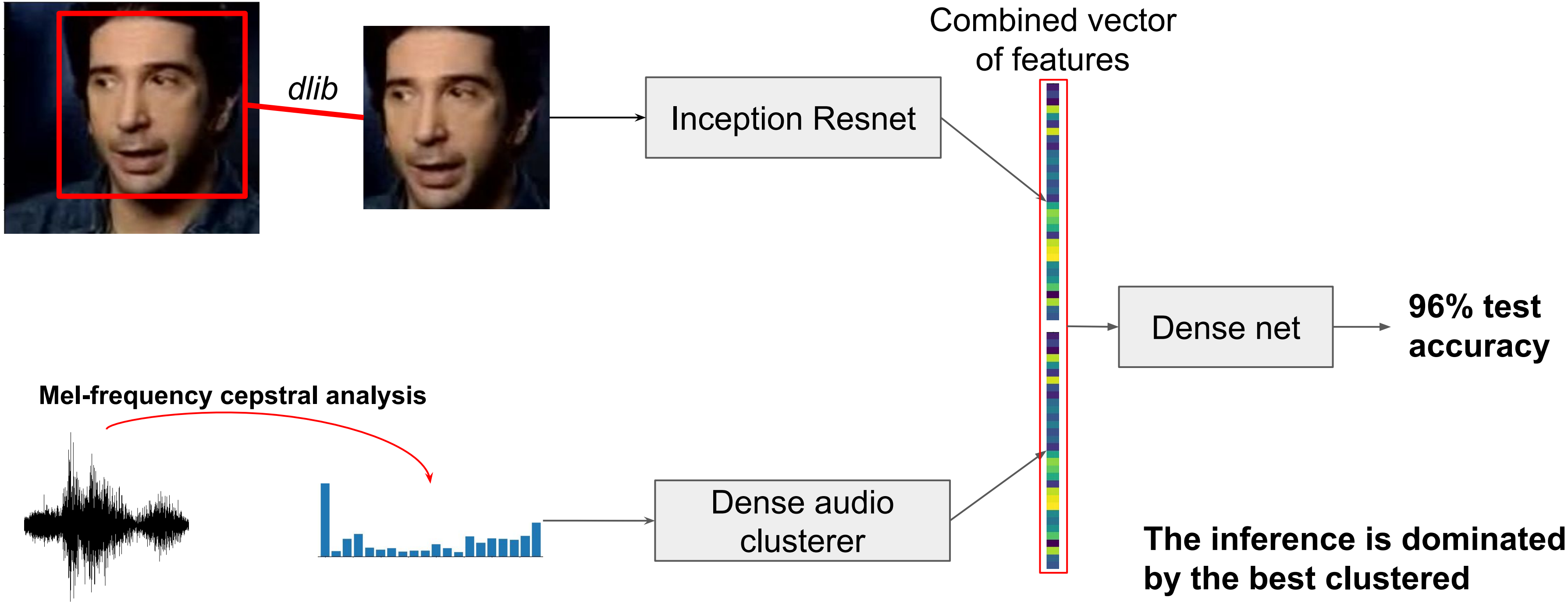
Resulting encodings

Dense net

Gets us to roughly 78.2% test accuracy

*$$D_{\text{positive}} - D_{\text{negative}} + 2$$

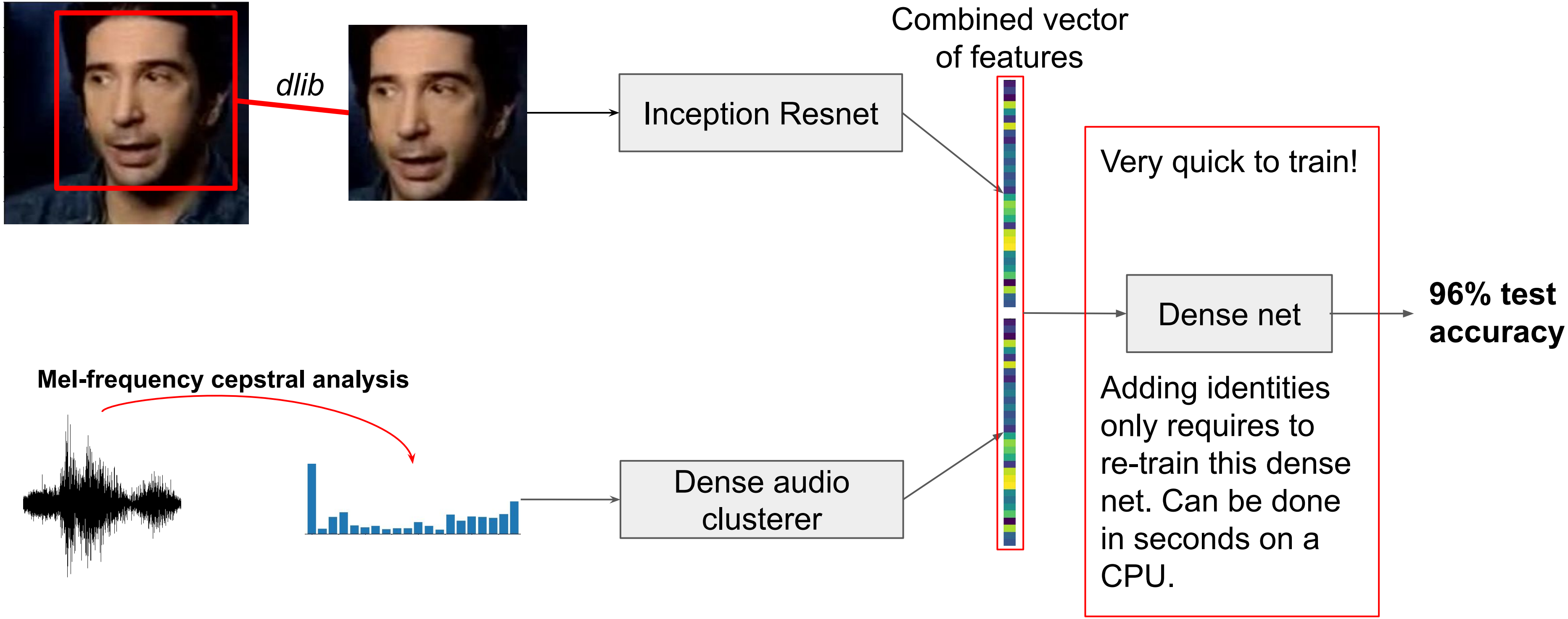
# General pipeline combining Audio & Images



**96% test accuracy**

**The inference is dominated by the best clustered feature set.**

# General pipeline combining Audio & Images



*dlib*

Inception Resnet

Combined vector of features

Mel-frequency cepstral analysis

Dense audio clusterer

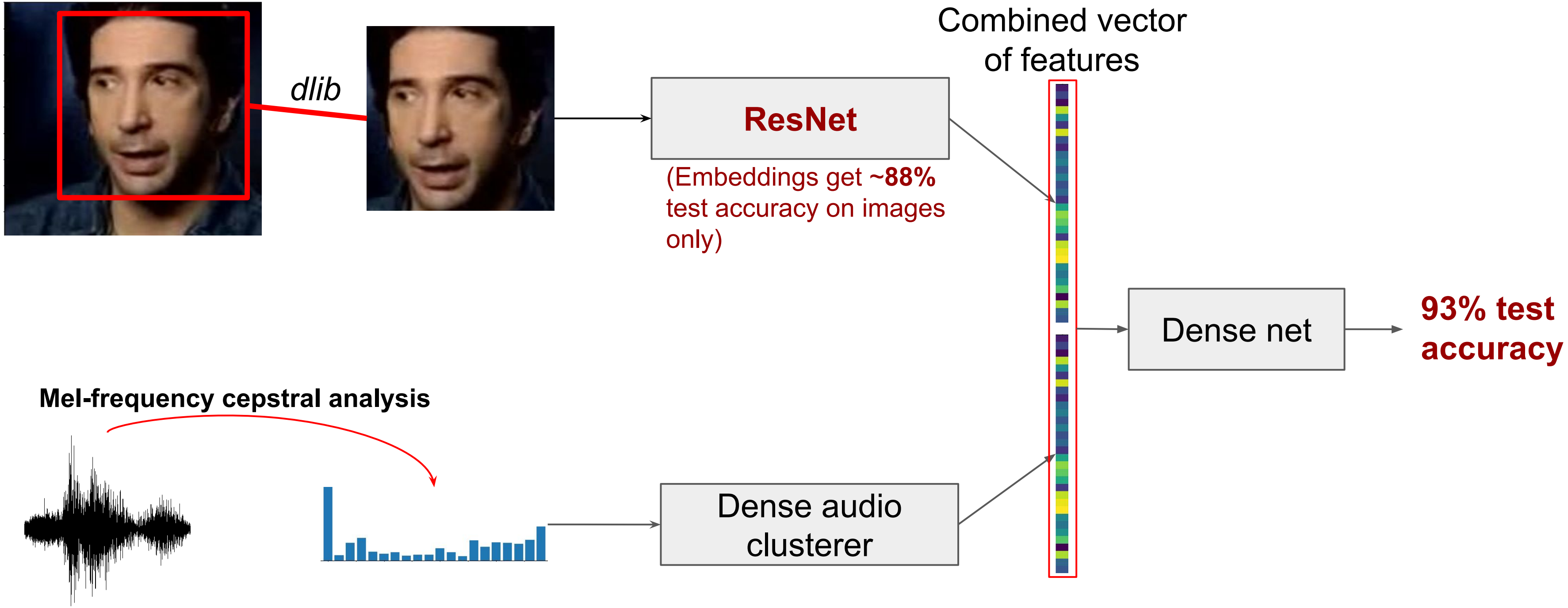
Very quick to train!

Dense net

Adding identities only requires to re-train this dense net. Can be done in seconds on a CPU.

**96% test accuracy**

# General pipeline combining Audio & Images



*dlib*

**ResNet**

(Embeddings get ~**88%** test accuracy on images only)

Combined vector of features

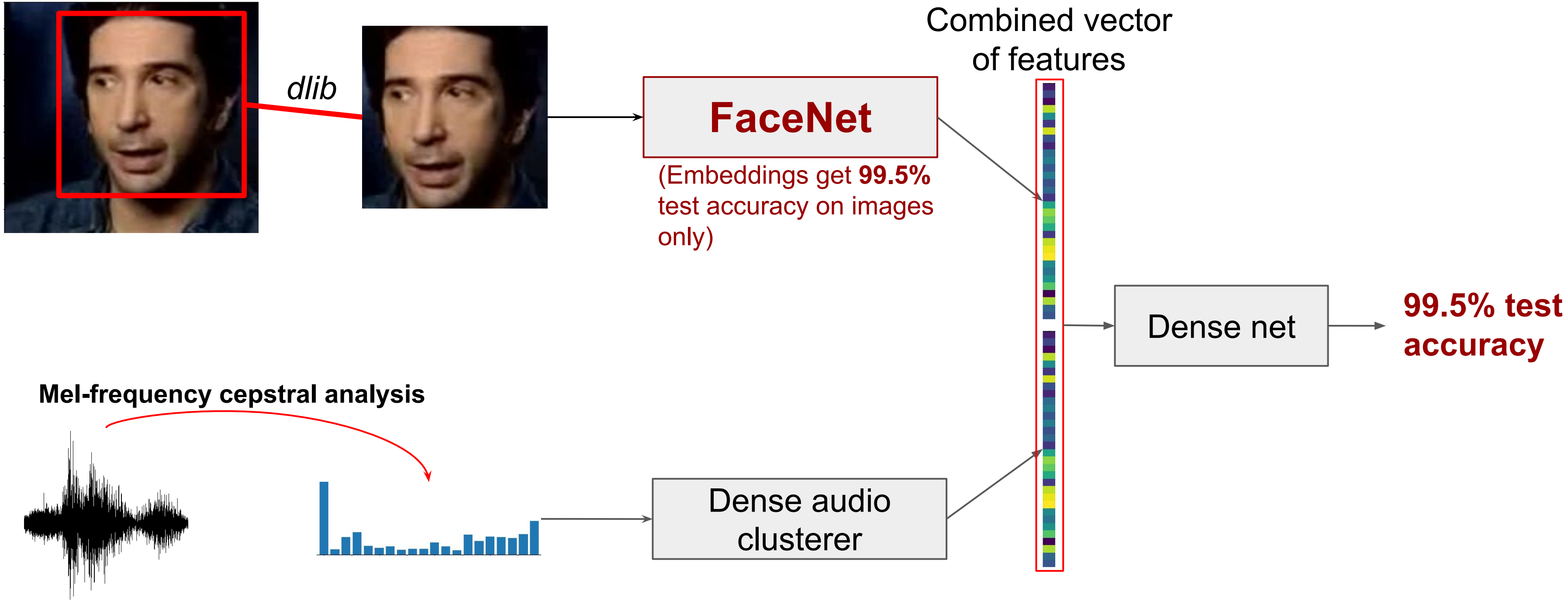
Dense net

**93% test accuracy**

**Mel-frequency cepstral analysis**

Dense audio clusterer

# General pipeline combining Audio & Images



*dlib*

**FaceNet**

(Embeddings get **99.5%** test accuracy on images only)

Combined vector of features

Dense net

**99.5% test accuracy**

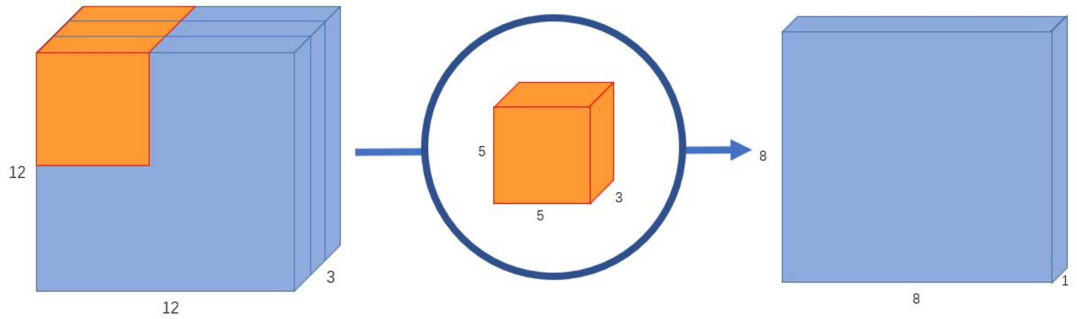
**Mel-frequency cepstral analysis**

Dense audio clusterer

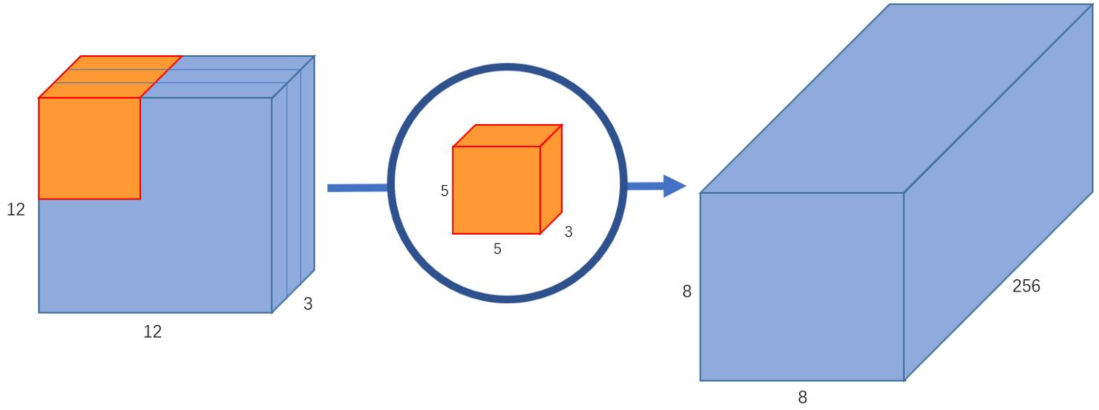
# Normal vs. Depthwise Separable Convolutions

# Normal Convolutions

1 Kernel: 5x5x3 Image: 12x12x3 -> 8x8x1



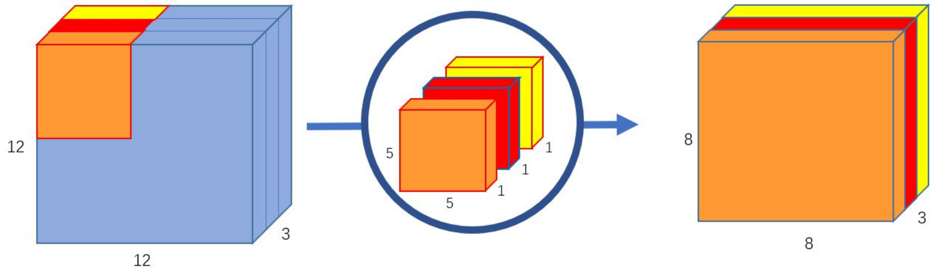
256 Kernels: 5x5x3 Image: 12x12x3 -> 8x8x1



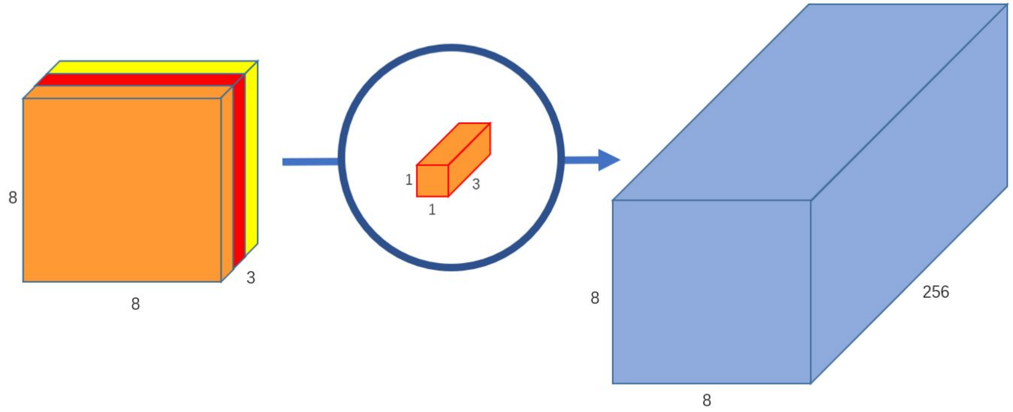
256x3x5x5x8x8 =1,228,800 multiplications

# Depthwise Separable Convolutions

**Step 1) Depthwise Convolution -** 3 Kernels (1 for each channel): 5x5x1 Image: 12x12x3 -> 8x8x3



**Step 2) Pointwise Convolution -** 256 Kernels: 1x1x3 Image: 8x8x3 -> 8x8x256



3x5x5x8x8 + 256x1x1x3x8x8 =
49,152 multiplications

(Reduction by a factor of 25)

# Image Classifier Comparison

| Network | Test Accuracy on Images | Combined Test Accuracy with Audio Classifier | No. of Parameters | Metric* |
|---|---|---|---|---|
| ResNet50 (with SeparableConv2D layers) | 90.1% | 93.7% | 13,618,884 | 92.34 |
| Truncated ResNet50 (by reducing no. of kernels) [with SeparableConv2D layers] | 89.3% | 93.3% | 8,539,972 | 92.45 |
| Truncated ResNet50 (by reducing no. of layers) [with SeparableConv2D layers] | 87.5% | 91.6% | 1,387,204 | 91.46 |
| Inception ResNet | 94.2% | 96.7% | 54,412,049 | 91.16 |
| Inception ResNet (with SeparableConv2D layers) | 94% | 96.6% | 40,606,444 | 92.54 |
| Truncated Inception ResNet (with SeparableConv2D layers) | 93.2% | 95.4% | 11,894,476 | 94.21 |

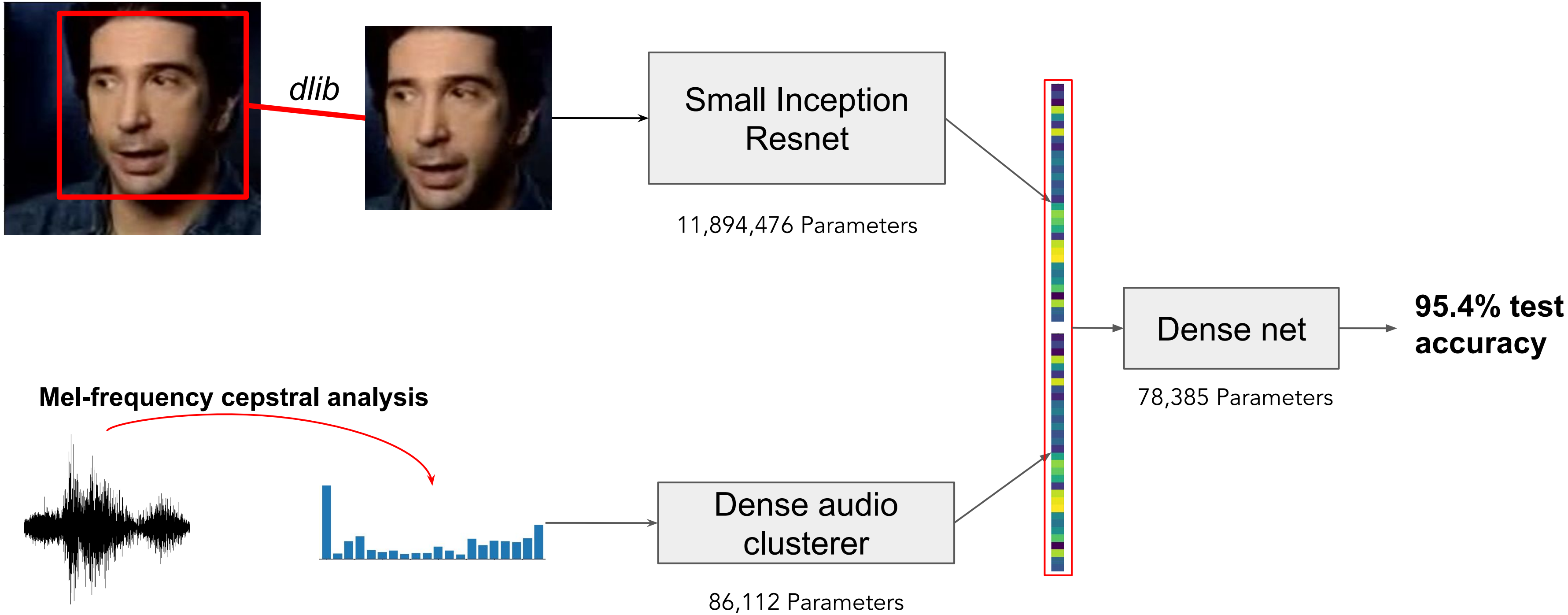
Our Audio Classifier has no Conv2D layers

*Metric = Accuracy - #Parameters / $10^7$

# Combined Classifier

2,866,334 equivalent parameters from Audio Pre-processing

1,187,412 equivalent parameters from Image Pre-processing



*dlib*

Small Inception Resnet

11,894,476 Parameters

**Mel-frequency cepstral analysis**

Dense audio clusterer

86,112 Parameters

Dense net

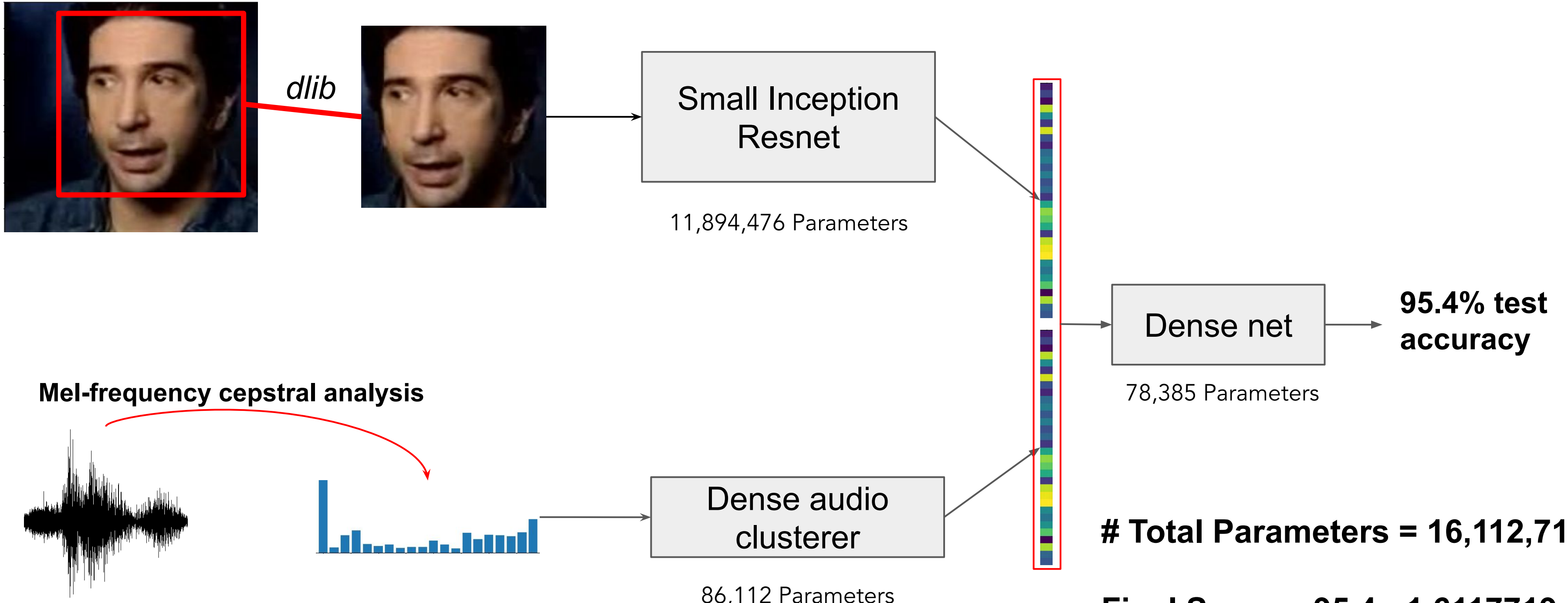
78,385 Parameters

**95.4% test accuracy**

# Combined Classifier

2,866,334 equivalent parameters from Audio Pre-processing

1,187,412 equivalent parameters from Image Pre-processing



*dlib*

Small Inception Resnet

11,894,476 Parameters

Dense net

78,385 Parameters

**95.4% test accuracy**

**Mel-frequency cepstral analysis**

Dense audio clusterer

86,112 Parameters

**# Total Parameters = 16,112,719**

**Final Score = 95.4 - 1.6117719**
**= 93.79**

# Thank You