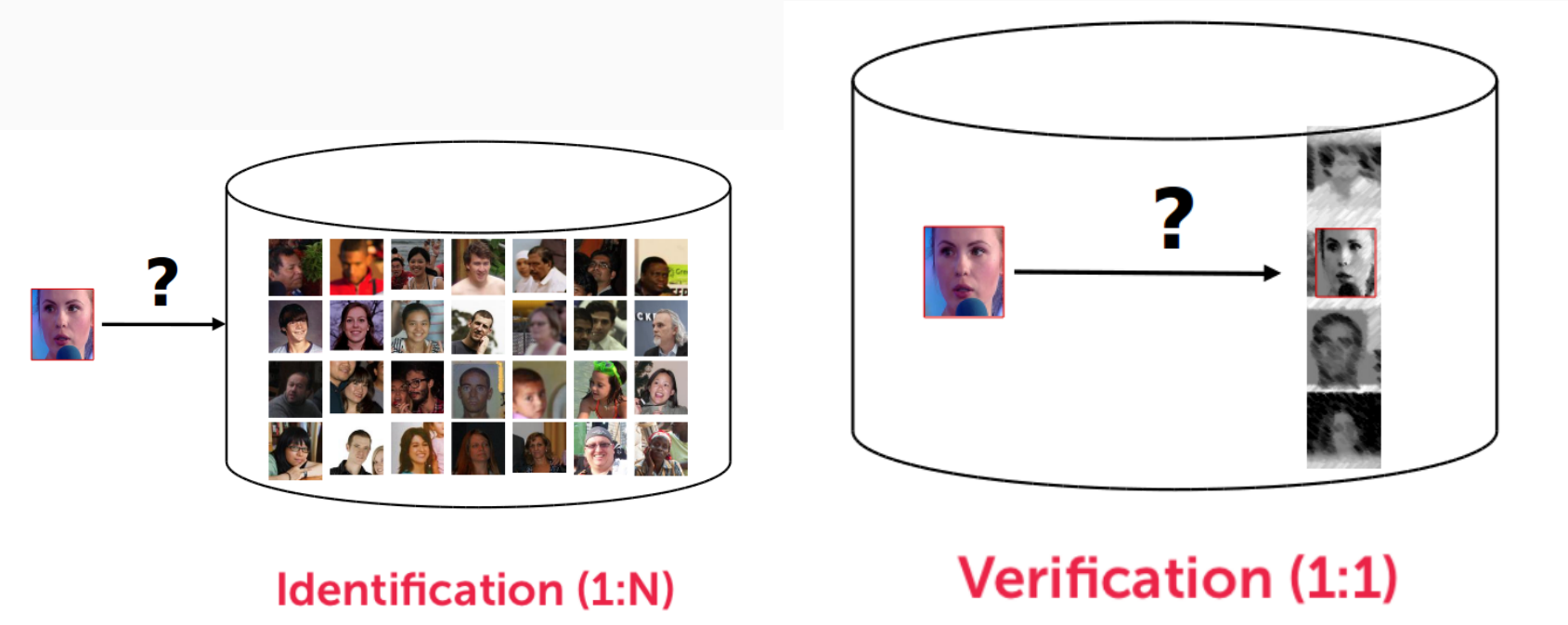
Face recognition at scale

Sergey Ovcharenko

Deep Learning Engineer @ NtechLab

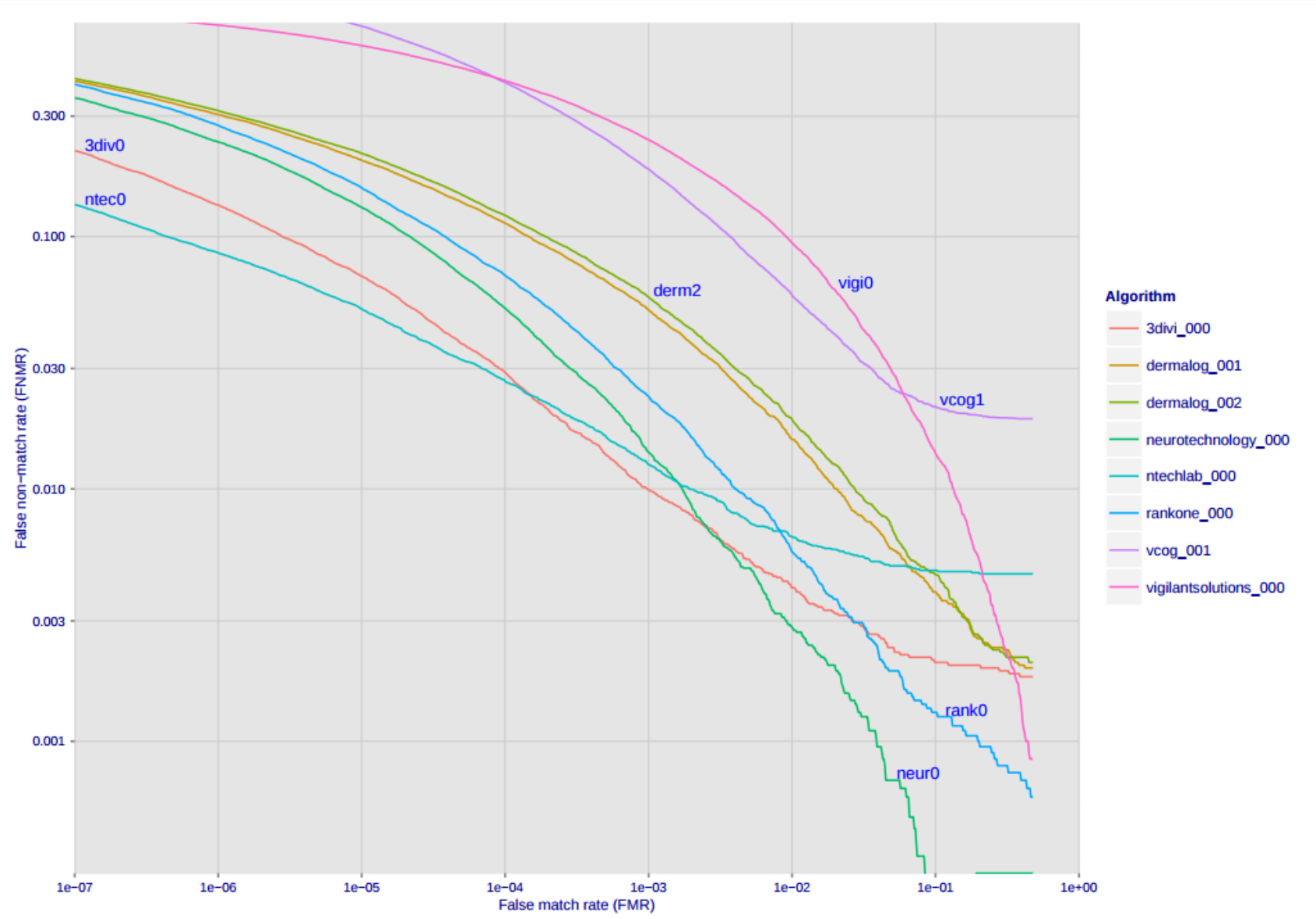
Face recognition tasks



Though related problems also arise e.g. (age, gender, emotion recognition, liveness detection, etc)

Face recognition challenges I

People are interested in extremely low FAR's:



Face recognition challenges II

You can’t get to know all persons so zero-shot scenario is required.

One needs to generalize well to unseen people, ethnicities, filming conditions.

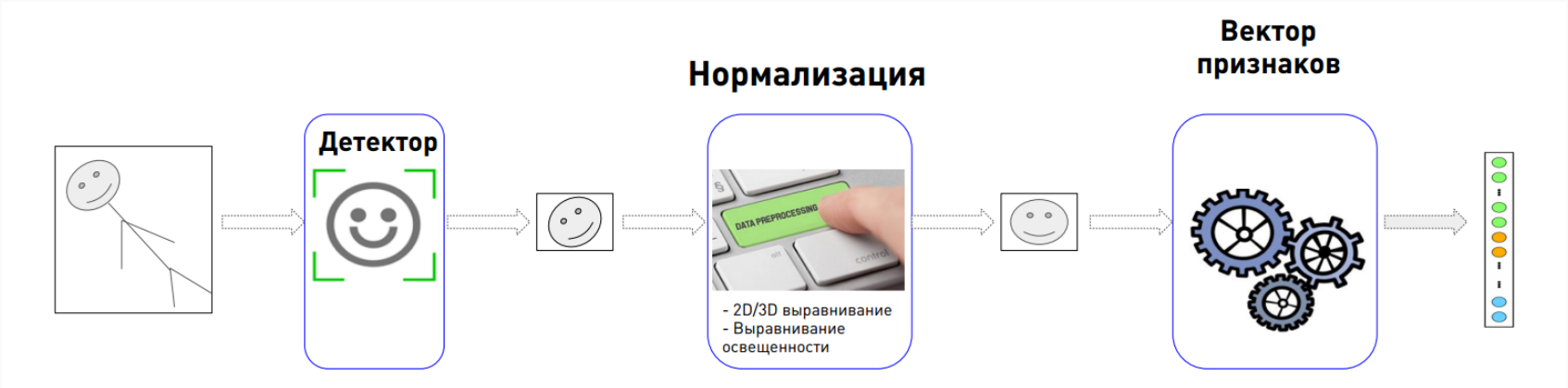
Face recognition challenges III

Your train set will most likely differ wildly from what you encounter in production:

- Class imbalance
- Crappy cameras
- Varying data origins
- Bad\no lighting

Are not your friends!

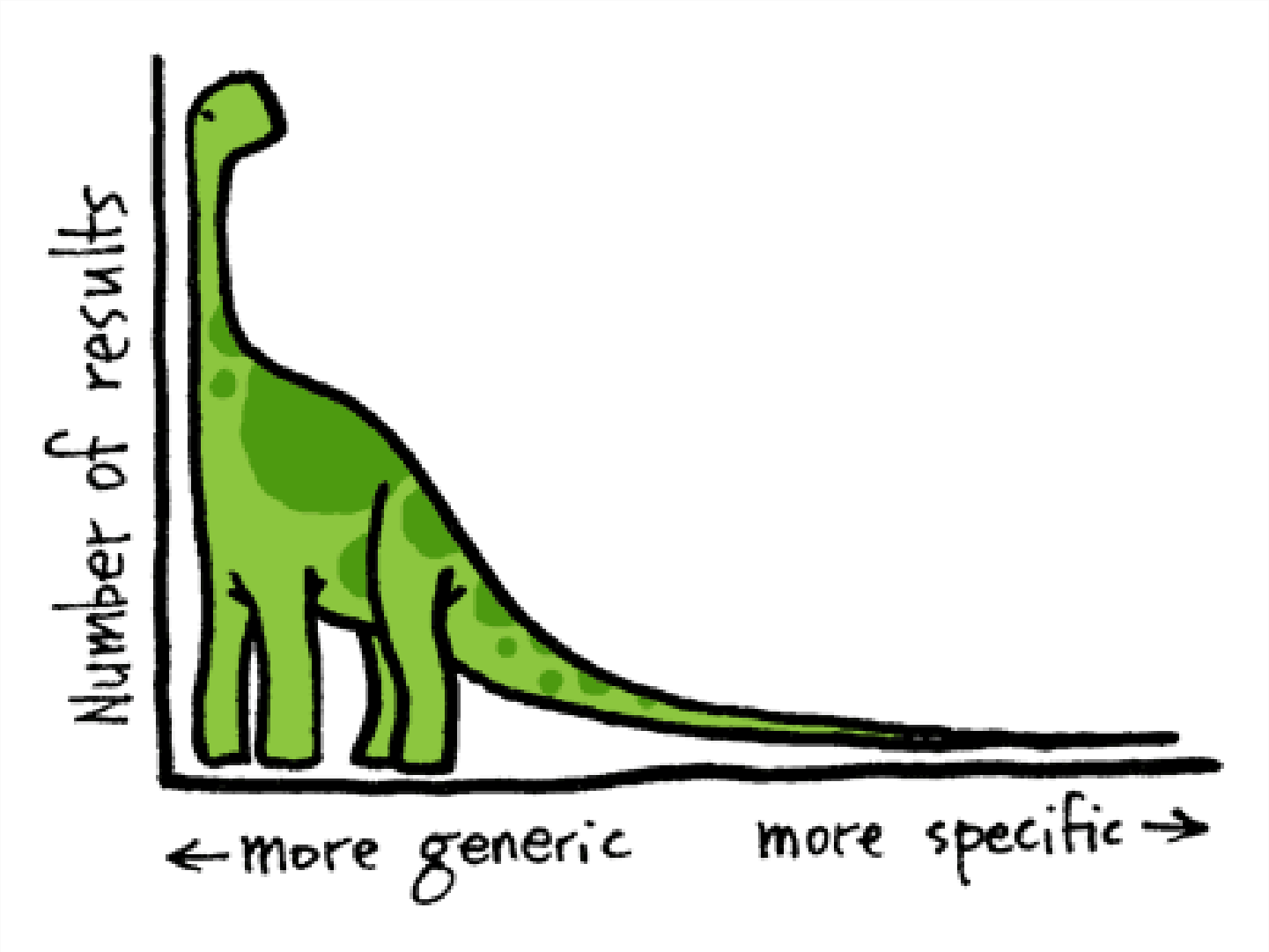
Pipeline


Детектор
Нормализация
DATA PREPROCESSING
- 2D/3D выравнивание
- Выравнивание освещенности
Вектор признаков

Detector

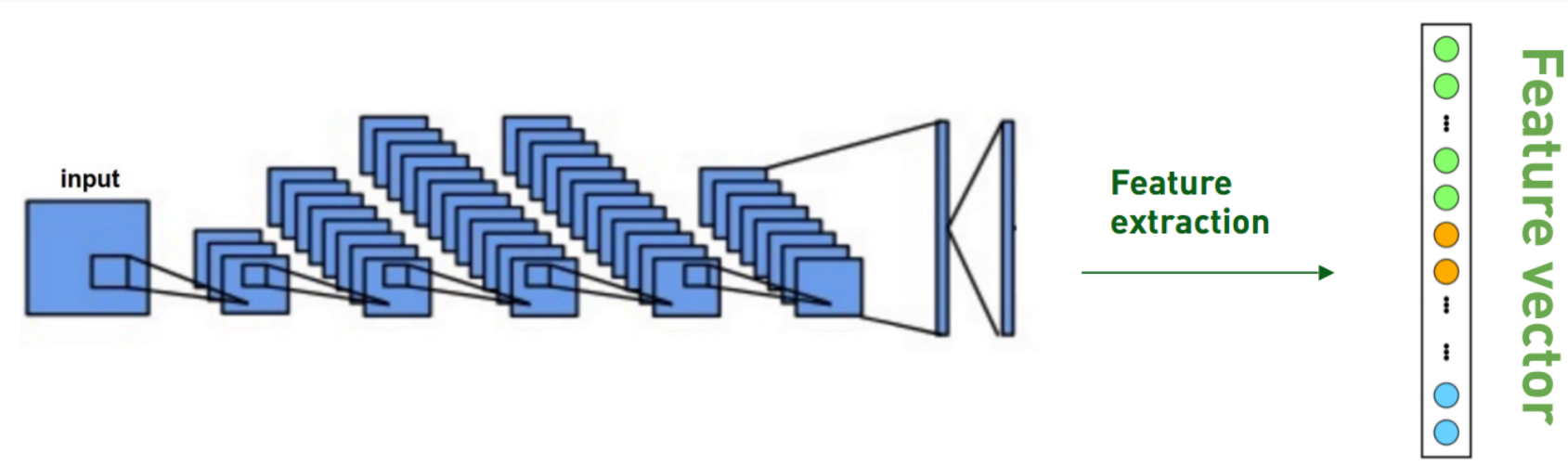
- Takes quite a while
- Should be fast and accurate
- We switched to CNN-based
- Also detects landmarks

CNN embeddings: dataset



Datasets are usually unbalanced and noisy

CNN embeddings



In essence the well-known bottleneck architecture with whistles and bells

CNN embeddings: approaches

Combine good feature extractor with specific loss functions:

Some popular choices:

- Contrastive loss:

$$L(X,\theta) = \frac{1}{2}\sum_{i=1}^{m} y_{ij}\left\|f(x_i) - f(x_j)\right\|_2^2 + (1-y_{ij})\left[\alpha - \left\|f(x_i) - f(x_j)\right\|_2\right]_+^2$$

- Triplet loss:

$$L(X,\theta) = \sum_{i=1}^{m}\left[\left\|f(x_i^a) - f(x_i^p)\right\|_2^2 - \left\|f(x_i^a) - f(x_i^n)\right\|_2^2 + \alpha\right]_+$$

- Center loss:

$$L_c(X,\theta) = \frac{1}{2}\sum_{i=1}^{m}\left\|\mathbf{x_i} - \mathbf{c_{y_i}}\right\|_2^2$$

- Others...

CNN embeddings: Practical considerations

As usually there's a trade-off between a most accurate model and what can be implemented in production.

A great deal of effort is put in obtaining compact representations.

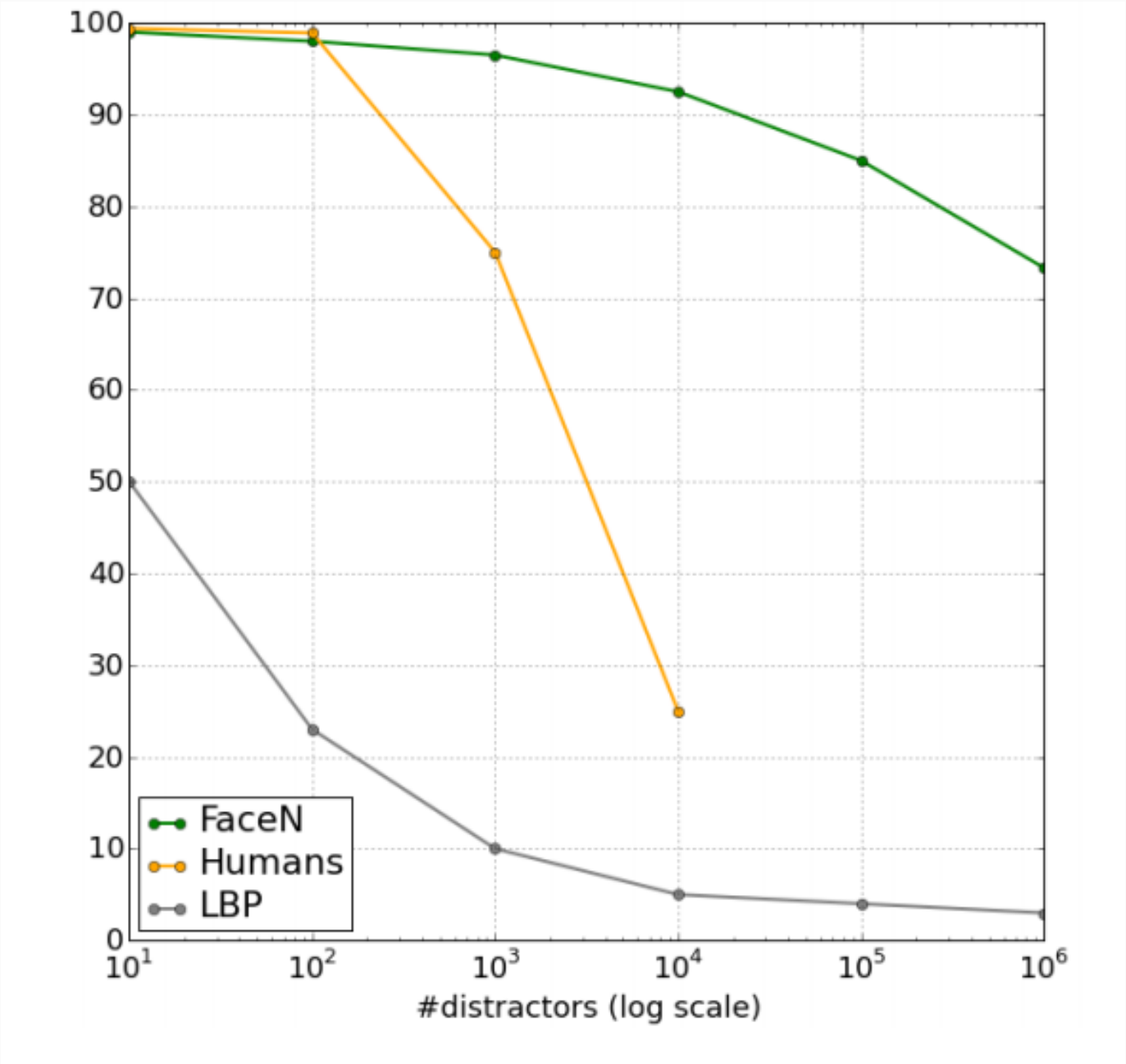
Binary embeddings can be very effective, but somewhat loose in terms of accuracy.

Nearest neighbors

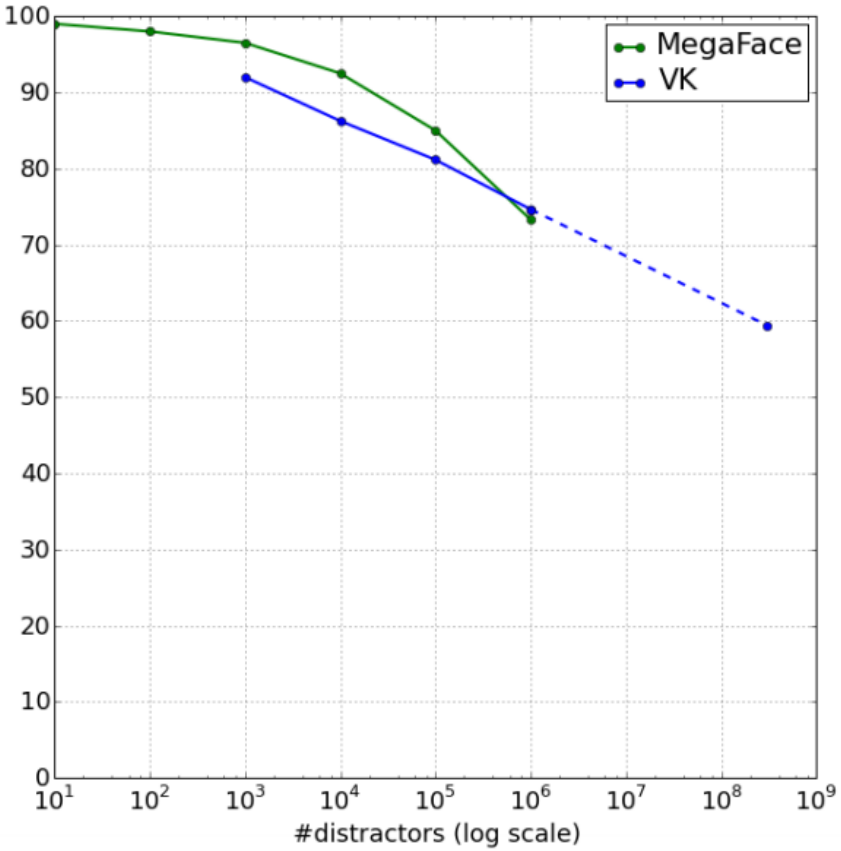
Once we have the embeddings, we need to search through them:

- Linear search
- Approximate nearest neighbors (e.g. projection trees)
- Custom fast binary index (hashing like)
- Cascading schemes

Goodies: CNN vs Humans



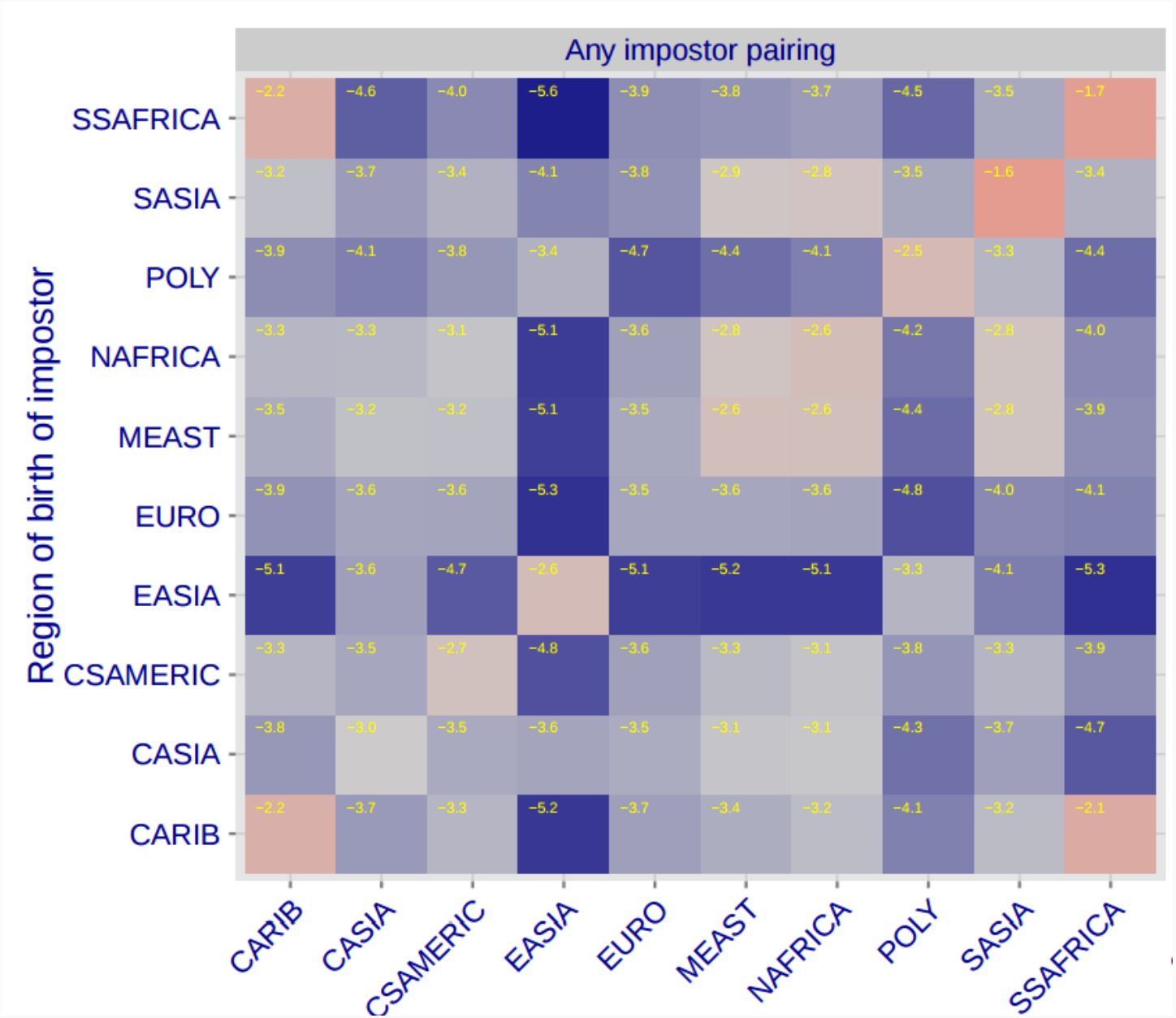
Goodies: MegaFace vs Internal



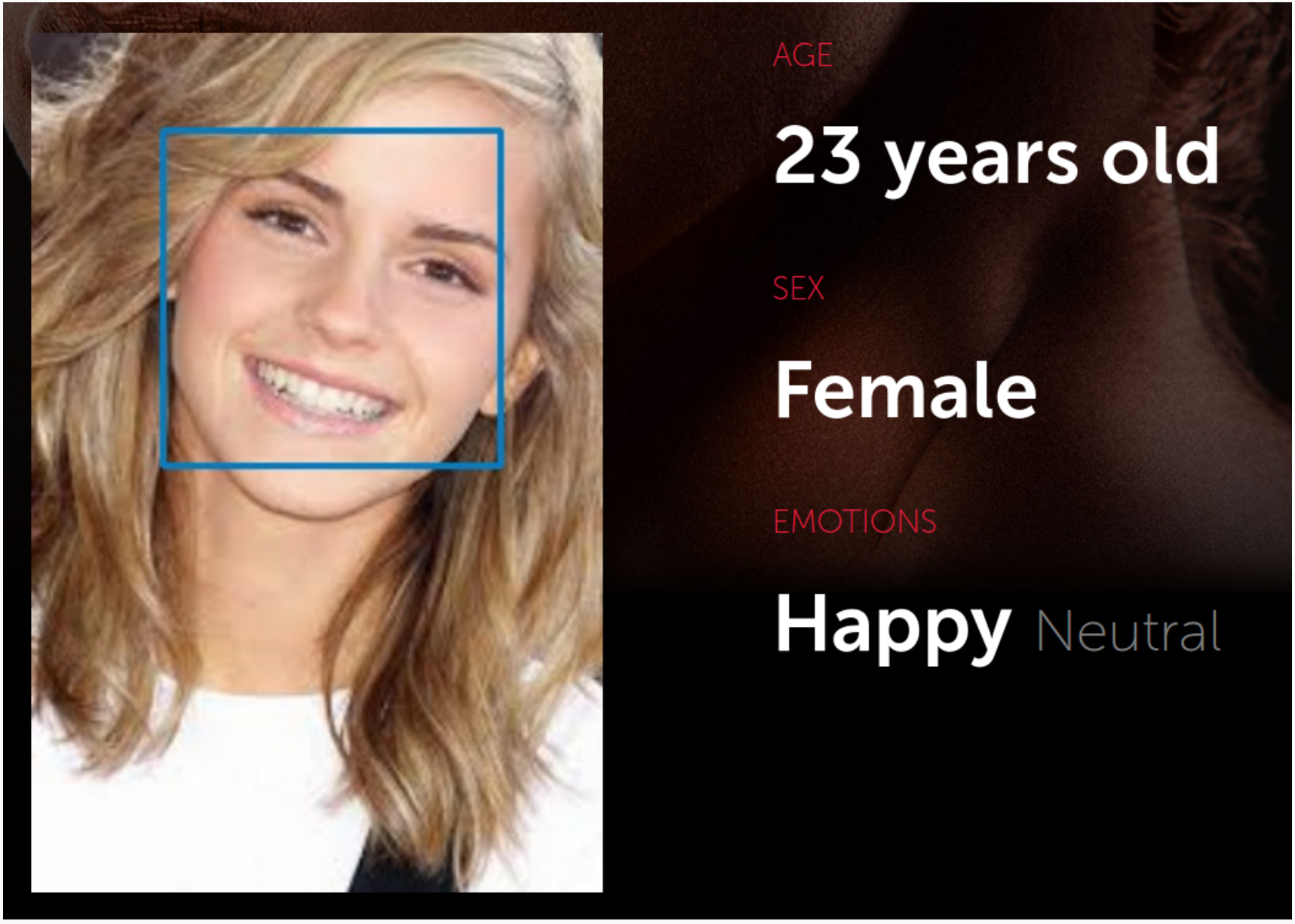
Rank-1 accuracy

Number of photos	Accuracy
1 M	73 %
250 M	60 %

Goodies: Impostor by region



Goodies: Age, gender and emotions



Frontiers: Liveness detection

Frontiers: Domain adaptation

Thank you

Thank you for your attention!