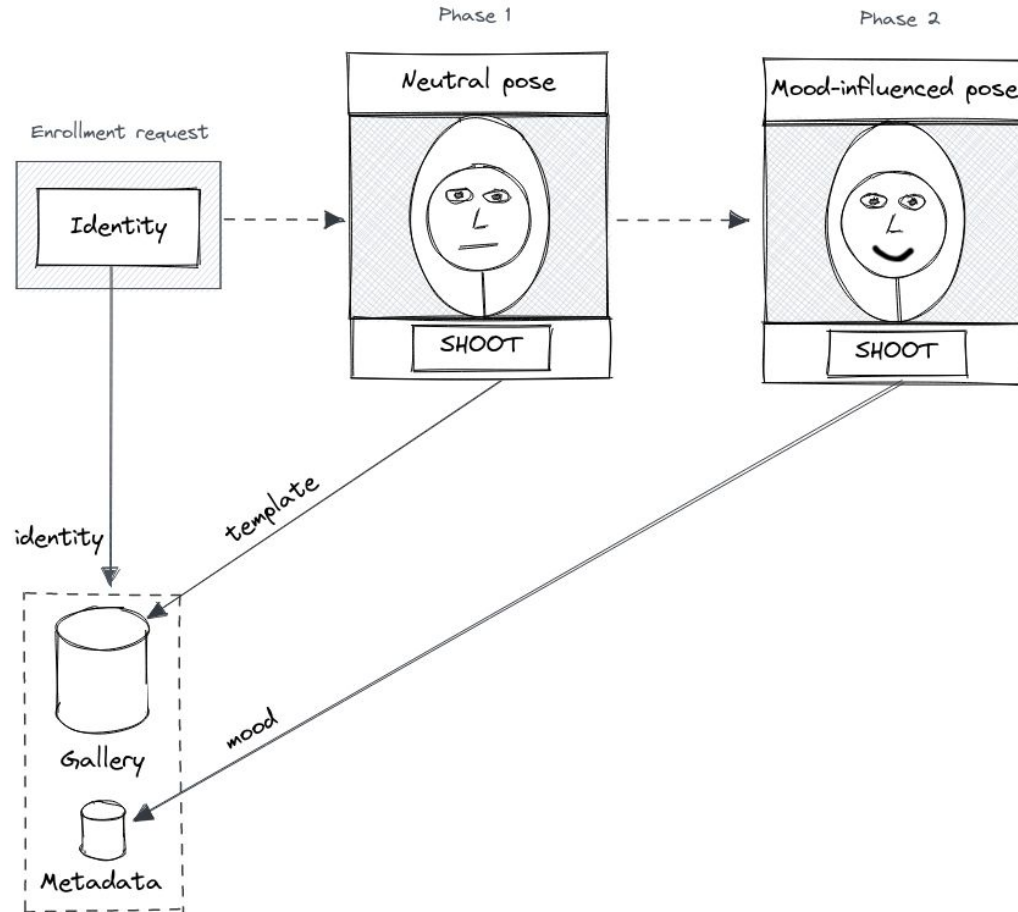


IdentiMood

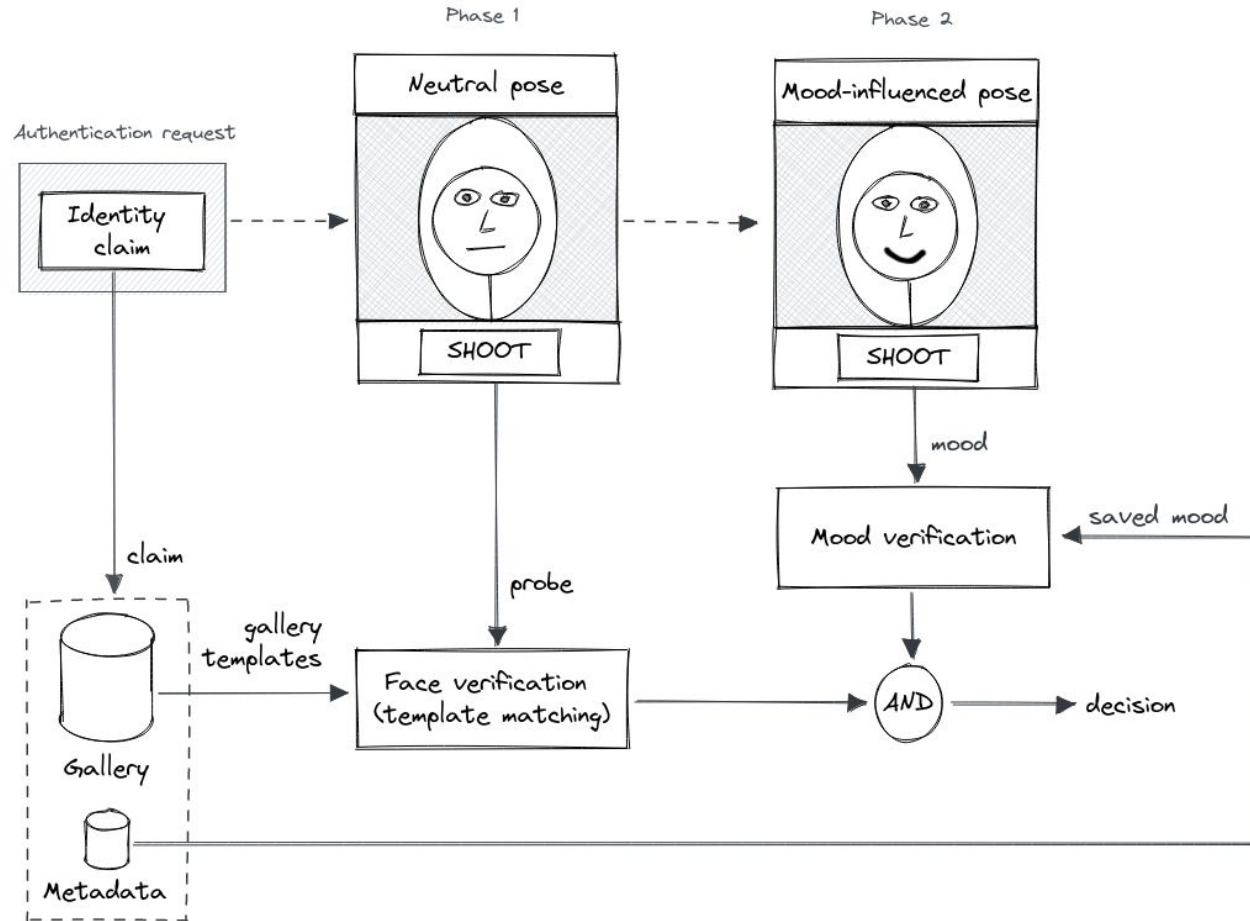
An authentication system based on
face verification and mood recognition

Daniele Solombrino, Davide Quaranta, Emanuele Volanti

Enrollment



Verification



Use case example

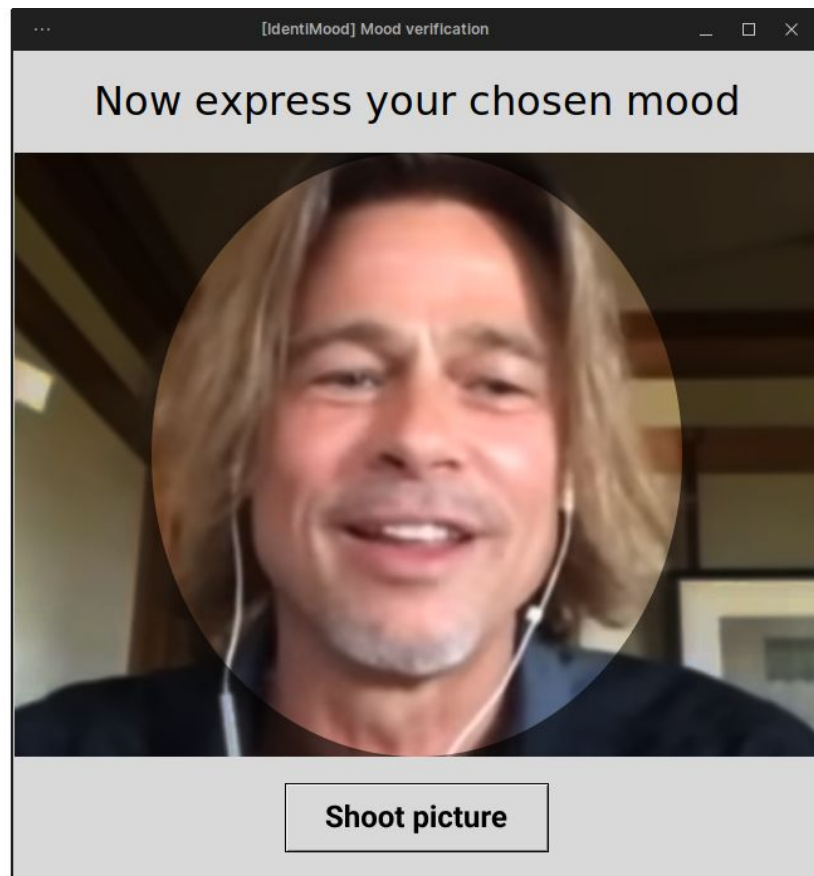
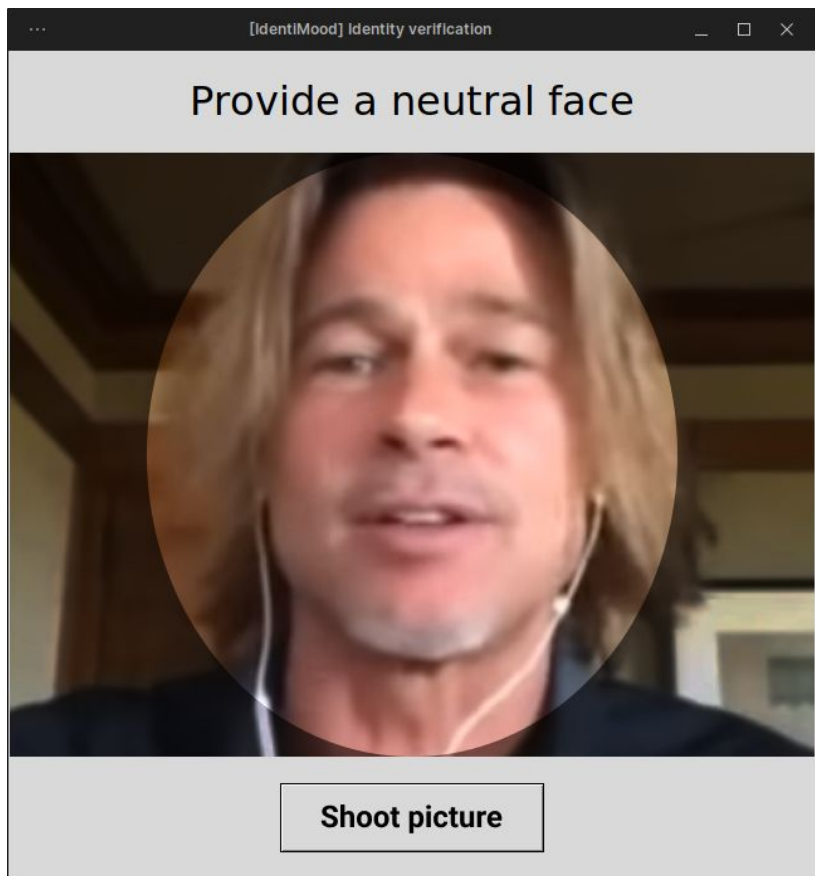


```
user@pc~ $ an-operation-that-triggers-identimood
```

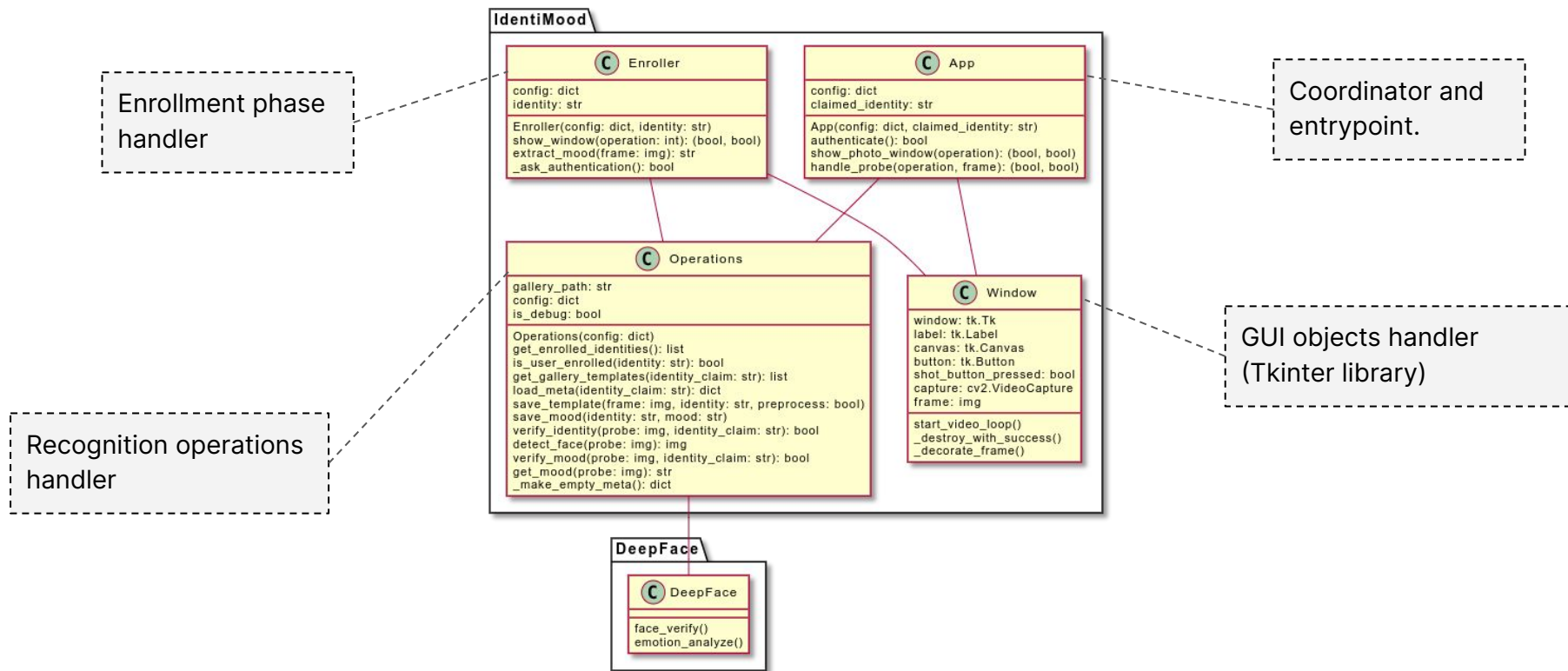
This operation requires authentication via IdentiMood.

Do you want to authenticate yourself?

[y]/n:



System design: low level view



System design: configurability

```
{
  "gallery_path": "./gallery",
  "face_verify": {
    "threshold": 0.12,
    "model_name": "Dlib",
    "detector_backend": "opencv",
    "distance_metric": "cosine",
    "normalization": "base",
    "save_normalized_templates": true
  },
  "mood_verify": {
    "use_delta_percent": true,
    "delta_percent_threshold": 70.0,
    "detector_backend": "opencv"
  },
  "debug": true
}
```

System design: gallery

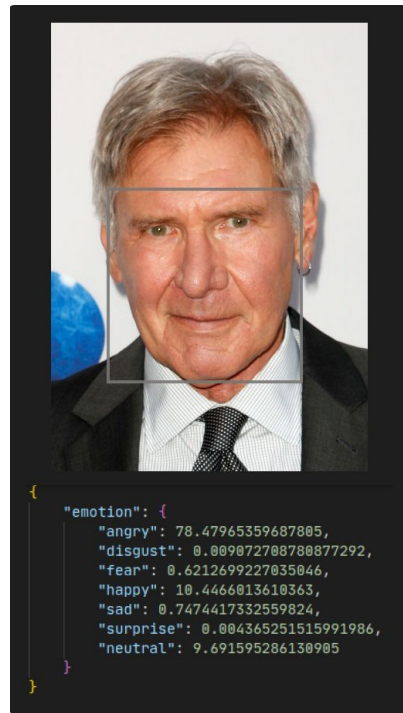
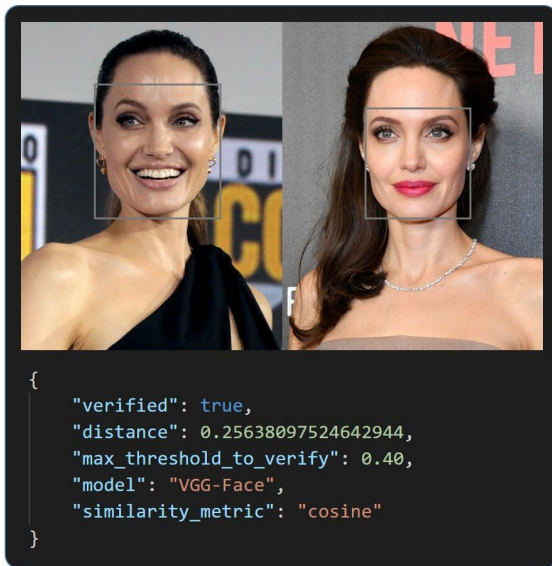
```
./gallery/  
├─ mary  
│   ├── 21eb2e26-a166-4e7e-92a8-065502e335bb.png  
│   └── meta.json  
└─ john  
    ├── 80a30386-cd18-4337-851a-9d2c965e07e3.png  
    ├── 82662f64-1188-46b8-b3c8-e895647c5dac.png  
    └── meta.json
```

```
{  
  "name": "Mary Smith",  
  "favorite_mood": "happy"  
}
```

- Directory-based
 - Each user → own directory
 - UUID → template name
- `meta.json` → user metadata

DeepFace (1/2): why?

- **Open Source** library
- Face **verification**
- **Mood** detection
- Preprocessing
- Flexible
- Pre-trained,
state-of-the-art
Deep Neural Networks



DeepFace (2/2): how?

Face **detectors**:

- OpenCV
- SSD
- Dlib
- MTCNN
- RetinaFace

Distance measures:

- Cosine
- Euclidean
- Euclidean with L2 norm

Face recognition **models**:

- VGG-Face
- OpenFace
- Facenet
- Facenet512
- DeepFace
- DeepID
- Dlib
- ArcFace

Evaluation: goal



- Face **detection**
 - Most reliable backend
- Face **verification**
 - Best combination of:
 - Model
 - Distance measure
 - Acceptance threshold
- **Mood** detection
 - Standard evaluation → evaluate DeepFace mood detection
 - “Delta-aware” evaluation → find the best delta

Evaluation: datasets

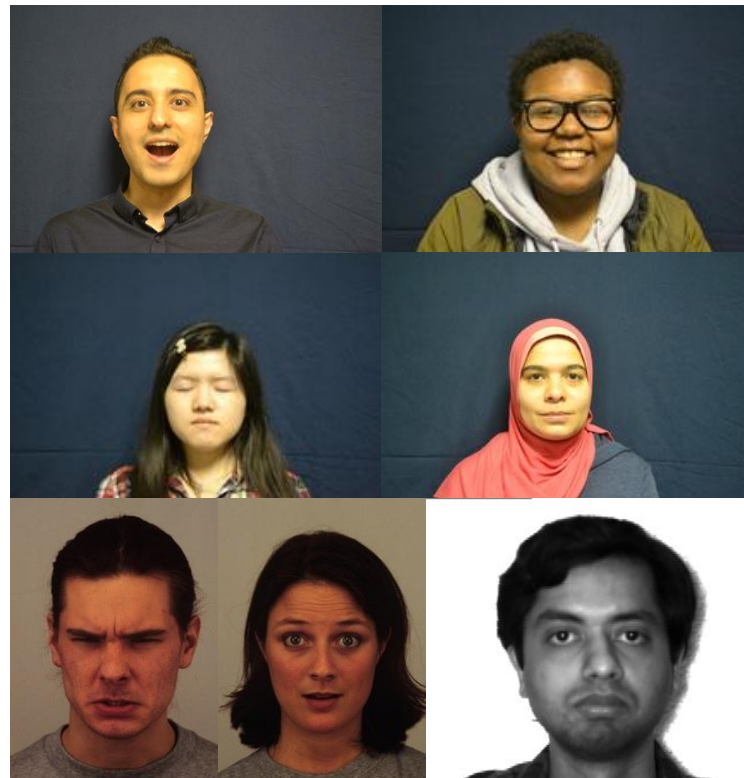
Desired characteristics:

- **Gender** variations
- **Ethnicity** variations
- **Age** variations
- **PIE** variations
- **Balancement**

Adopted datasets:

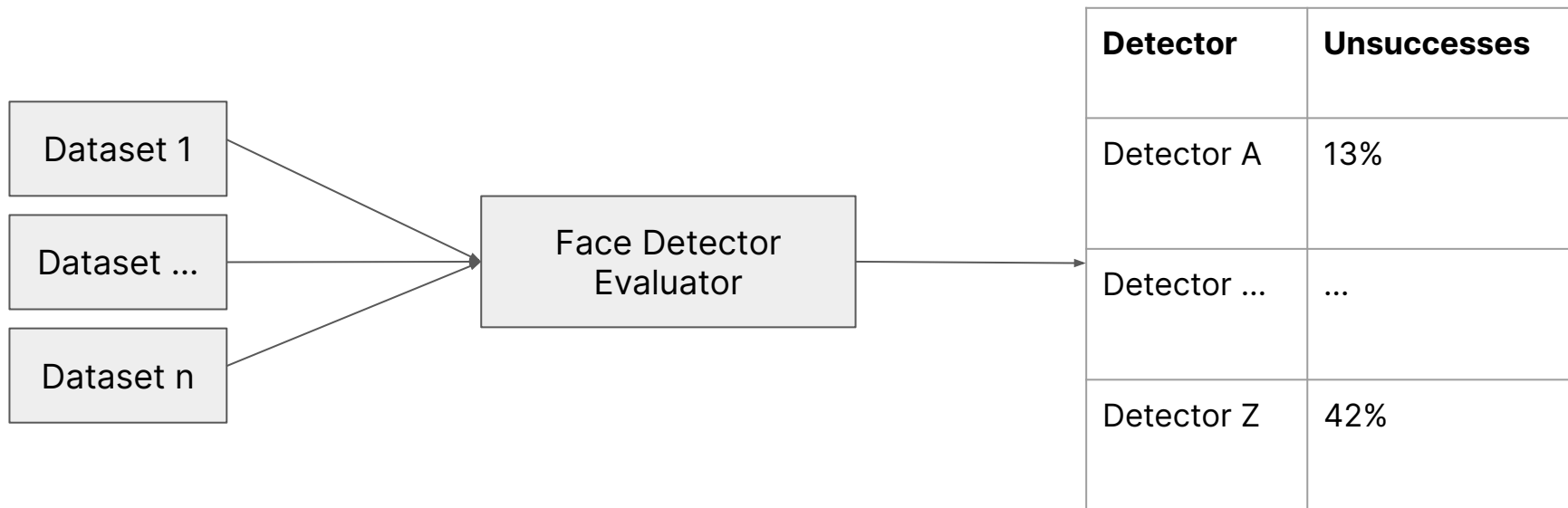
TUTFS, KDEF, YaleFaces, VGG-Face2

~130k images



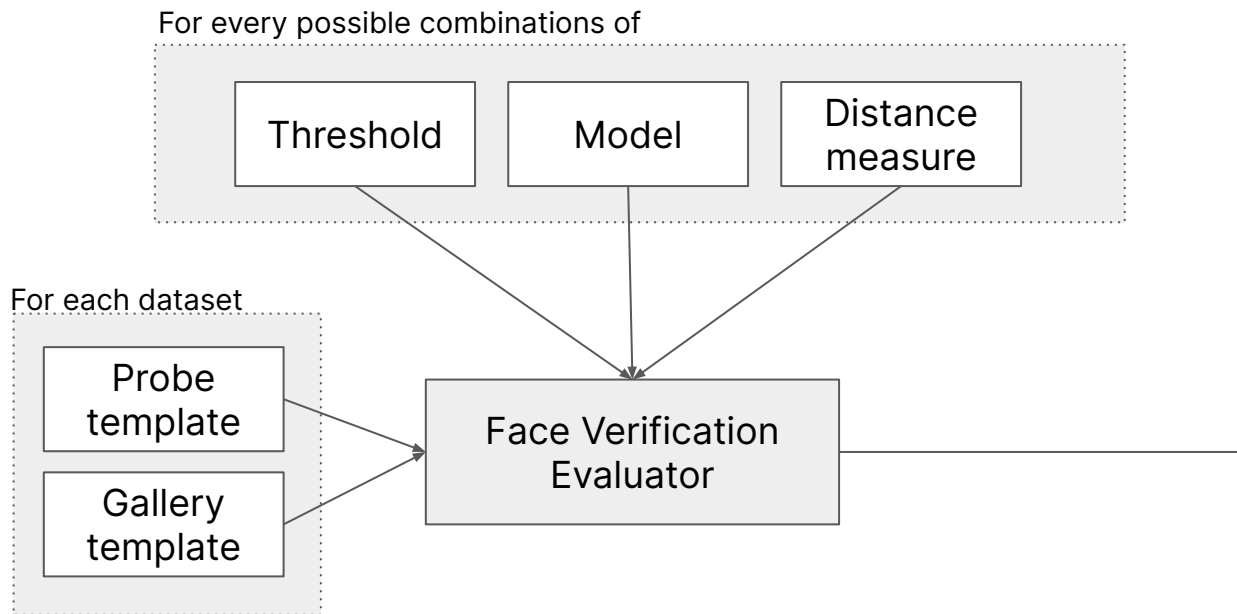
Evaluation: face detector

- Find **most reliable** backend
- Stick to it throughout the evaluation process



Evaluation: face verification (1/2)

All-against-all distance matrix



Evaluation: face verification (2/2)

Dataset	Model	Distance measure	Threshold	FRR	FAR
...
...
...

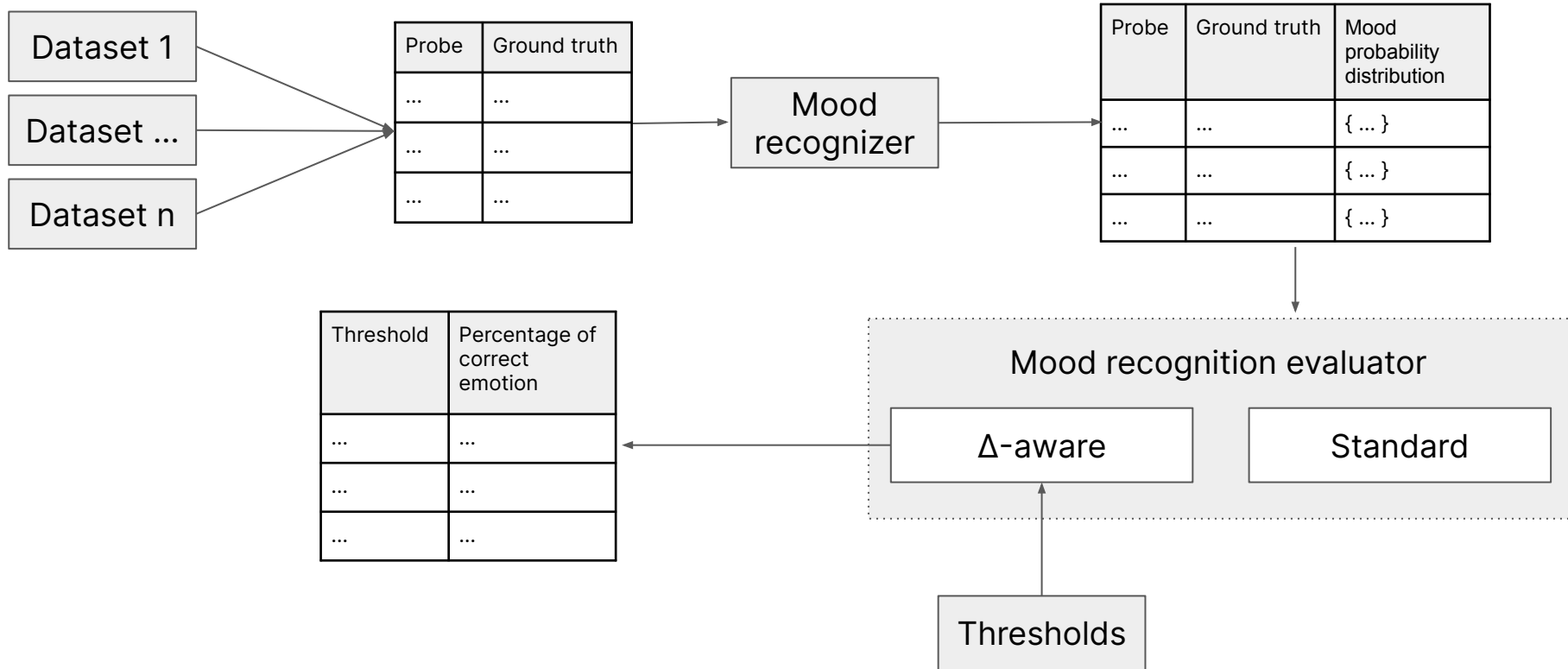
Figures of merit
calculator

ROC

DET

Thresholds vs
FAR and FRR

Evaluation: mood recognition



Δ -aware emotion verification



Δ = 1st emotion ranking - 2nd emotion ranking

Consider emotion as **correctly detected** if:

- 1st ranking emotion = ground truth
- $\Delta \geq \text{threshold}$

Evaluation execution

Multiple compute-oriented VMs:

- Parallelization by configuration
- Workload distribution



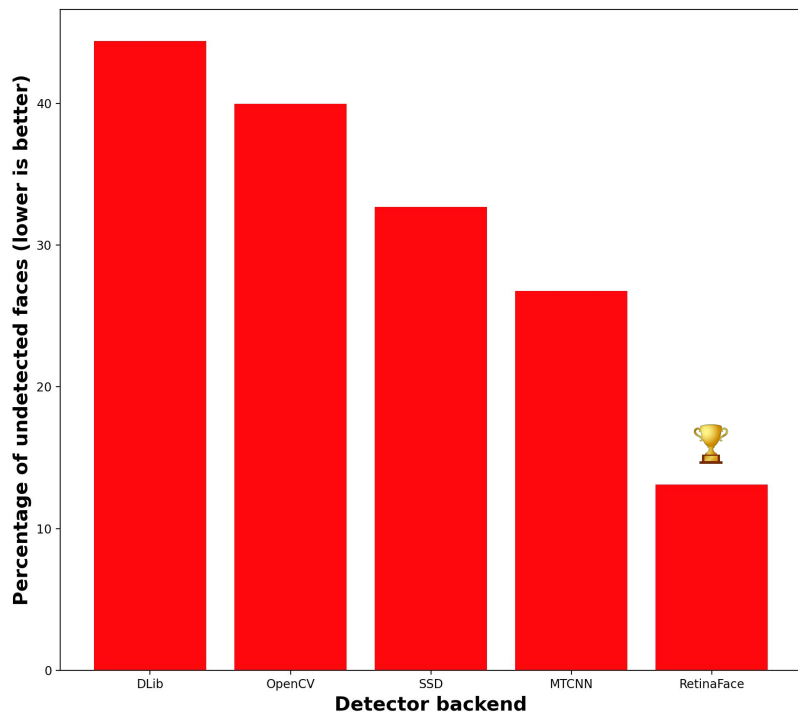
```
$python3 all_against_all.py -i dataset_files.txt -l 100 -dc -de -del2 -tr 0 1 100 -m 11100000 -v
```

Diagram illustrating the command-line arguments for the evaluation script, with labels for each argument:

- `-i dataset_files.txt`: File list
- `-l 100`: Limit gallery and probe size to 100
- `-dc`: Cosine
- `-de`: Euclidean
- `-del2`: Euclidean L2
- `-tr 0 1 100`: Linear space to generate threshold list
- `-m 11100000`: Models to use
- `-v`: Verbose

Evaluation results: detector backends

Face detection errors for each backend detector



RetinaFace → fewest no-detections 🏆

Dlib → most no-detections

Evaluation results: face verification

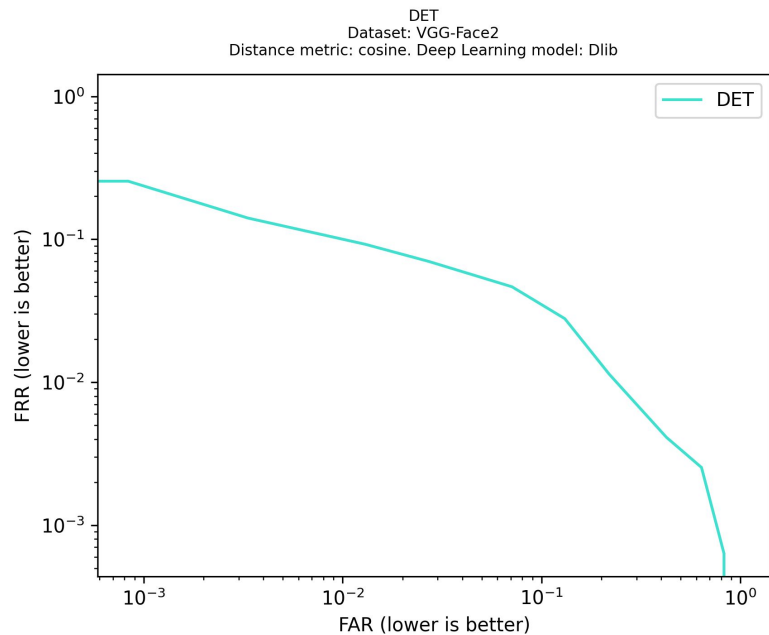
YaleFaces	Best			Worst		
	DET	ROC	Thresholds vs FAR and FRR	DET	ROC	Thresholds vs FAR and FRR
Cosine	Facenet512	VGG-face	Facenet512	DeepID	DeepID	DeepID
Euclidean	Facenet512	Facenet512	Facenet512	DeepID	DeepID	DeepID
Euclidean L2	Facenet512	Facenet512	Facenet512	DeepID	DeepID	DeepID
VGG-Face2	Best			Worst		
	DET	ROC	Thresholds vs FAR and FRR	DET	ROC	Thresholds vs FAR and FRR
Cosine	Dlib	Dlib	Dlib	DeepID	DeepID	DeepFace
Euclidean	Facenet512	Facenet512	Facenet512	DeepID	DeepID	DeepID
Euclidean L2	Facenet	Dlib	Dlib	DeepID	DeepID	DeepID
...		

Plots (300+) for each dataset, model, metrics and threshold combination
(2.5+ mln)

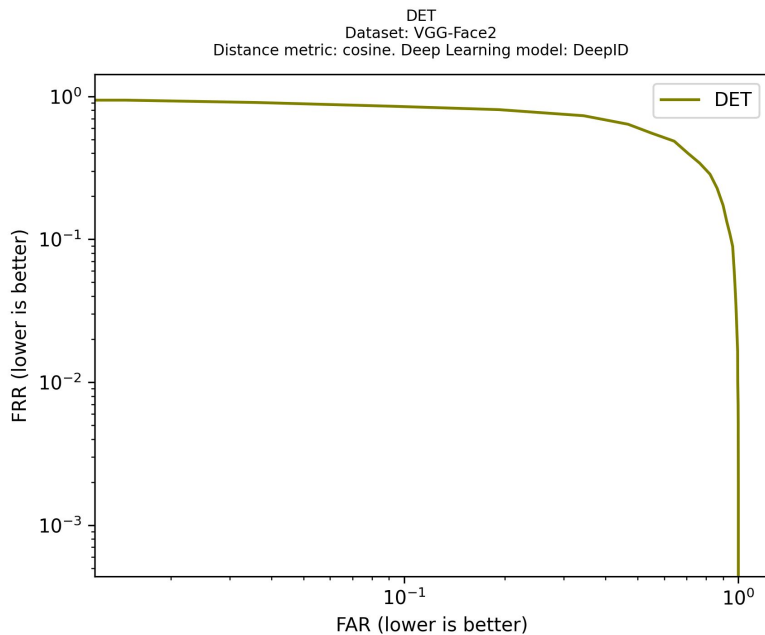
DET

(Lowest AUC is better)

Best: Dlib



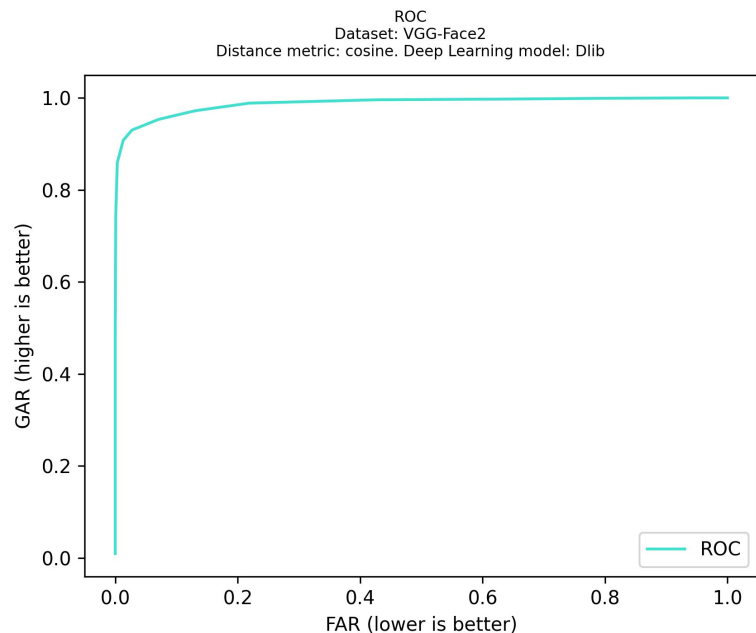
Worst: DeepID



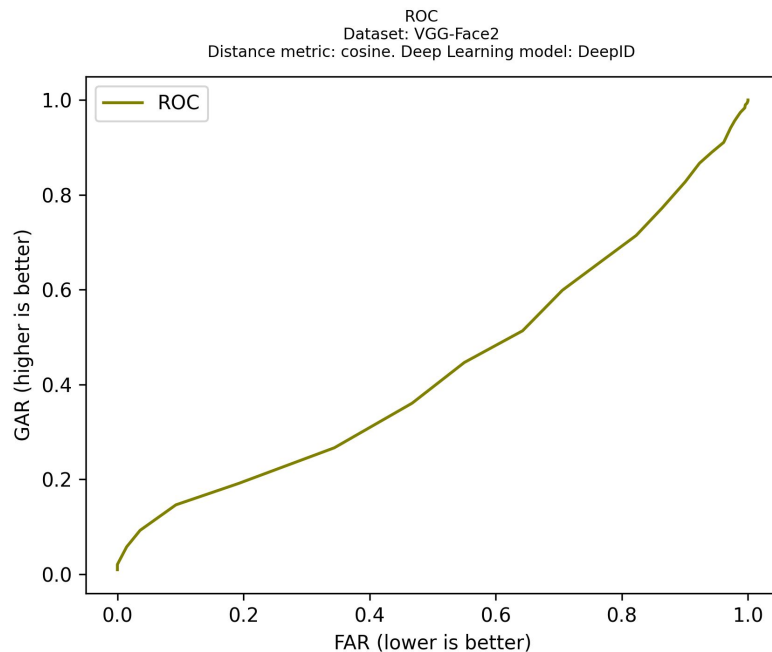
ROC

(Highest AUC is better)

Best: Dlib



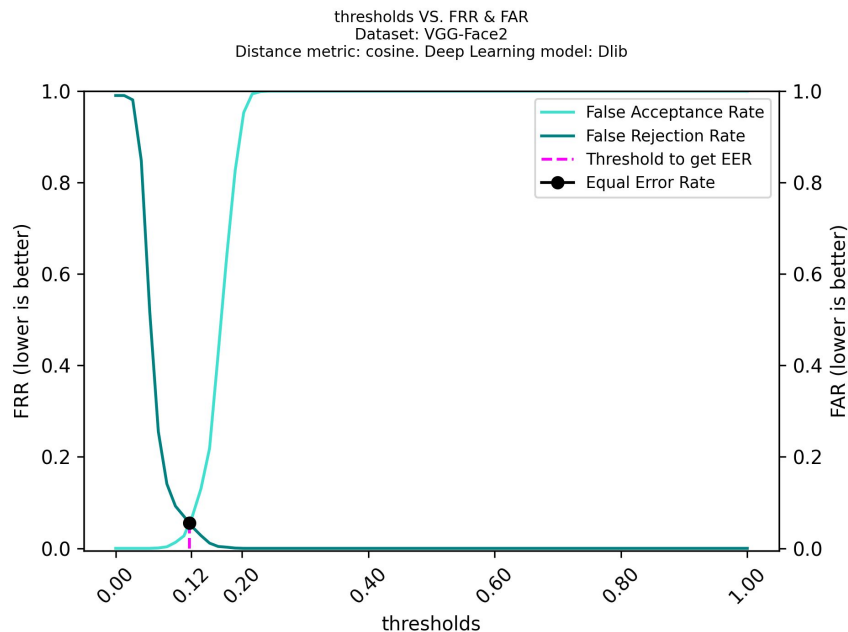
Worst: DeepID



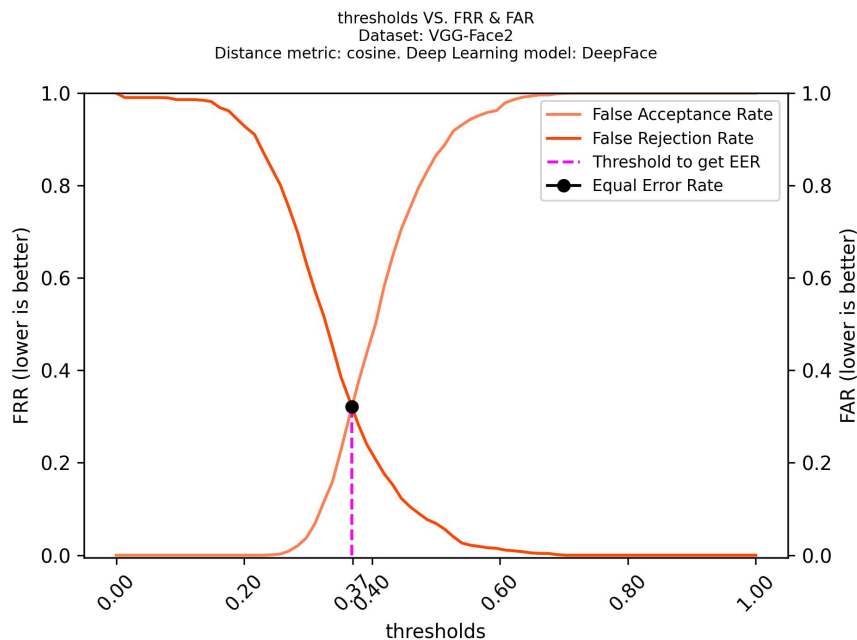
Thresholds vs FAR & FRR

(Lowest EER is better)

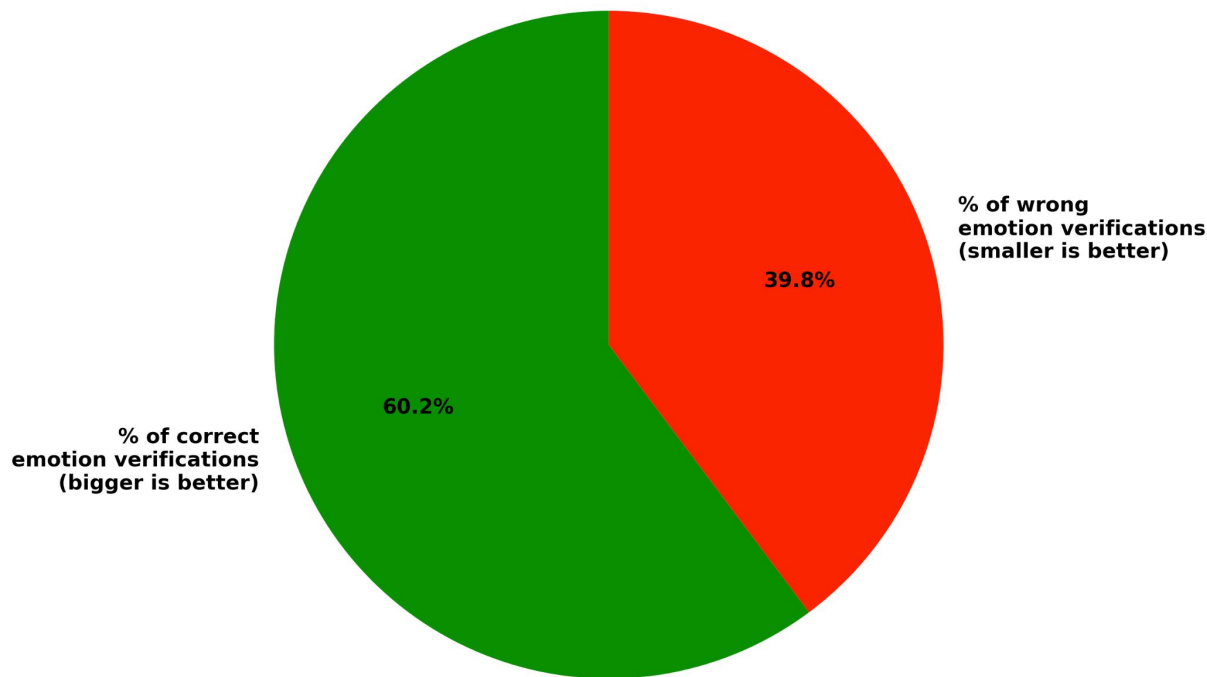
Best: Dlib



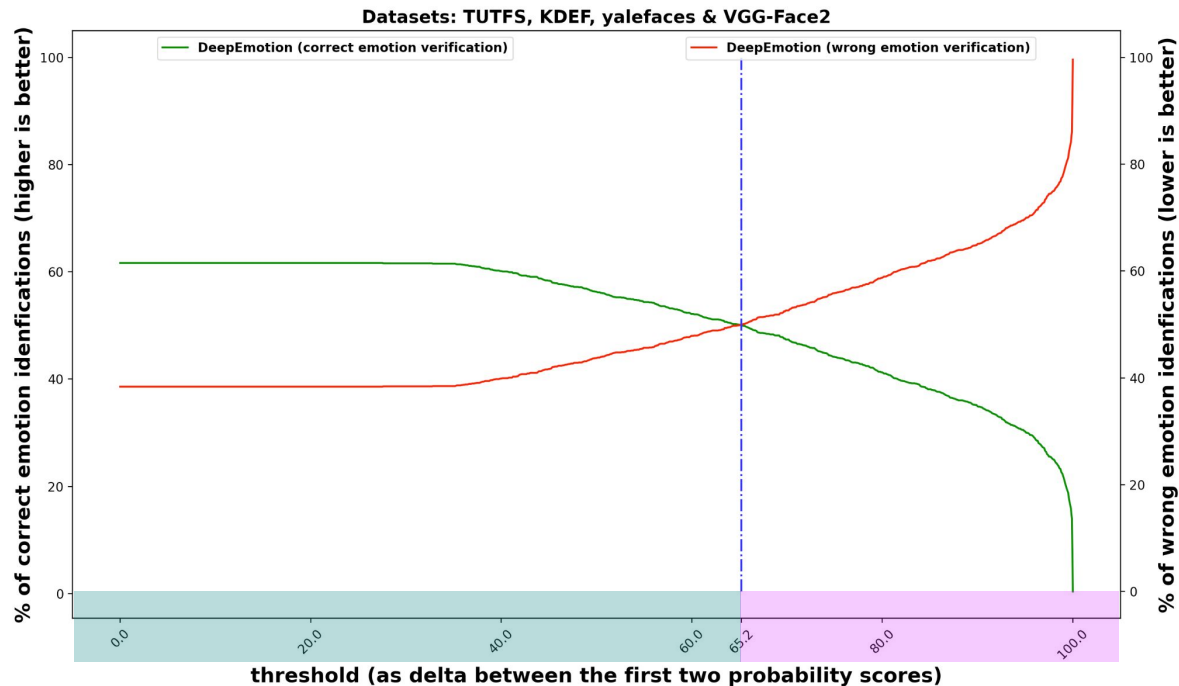
Worst: DeepFace



Evaluation results: standard emotion verification



Evaluation results: Δ -aware emotion verification



Better than random
Same as random
Worse than random

Conclusion

- Authentication system
- **Face** verification
- **Mood** recognition
- Python3 application
- **DeepFace** library
- Evaluation of every module:
 - **ROC, DET, Thresholds** vs **FAR** and **FRR** → face verification
 - Delta-aware verification → **emotion** recognition
 - Multiple runs on cloud **VMs**
 - 300+ plots produced → **every** possible **combination** evaluated