академия больших данных

mail.ru group

ML System design

Эдуард Тянтов

Руководитель направления машинного обучения
в Почте и Портале

Plan

- Steps of ML System Design
- 2 Case studies
  - OCR @Scale
  - Face recognition on video

Steps of System Design

Steps

1. Problem statement
2. Data
3. MVP
4. Model
5. Metrics
6. Deployment
7. System design

1. Problem statement

1. Clear objective
    1. Business, Error budget
    2. ML, FP vs FN
2. Scale
3. Inference budget & other restrictions

2. Data

1. Data sources
2. Supervision signal
3. Feedback loop

3. MVP

mail.ru group

Goal – to learn as much as possible

1. Viability
2. Open questions
3. Potential problems/risks
4. New restrictions

4. Model

- Feature engineering
  - connections between different types of data
  - hand-crafted/automated/statistics
- Model
  - Restrictions
  - Simplicity
  - Explainability
  - …
- Architecture, losses, training, …

5. Metrics

Specify target metrics

6. Deployment

- A/B testing
- Monitoring
- Framework

7. System design*

Infrastructure around the ML component:

- Databases
- Scalability
- Availability
- Security
- Trade-offs

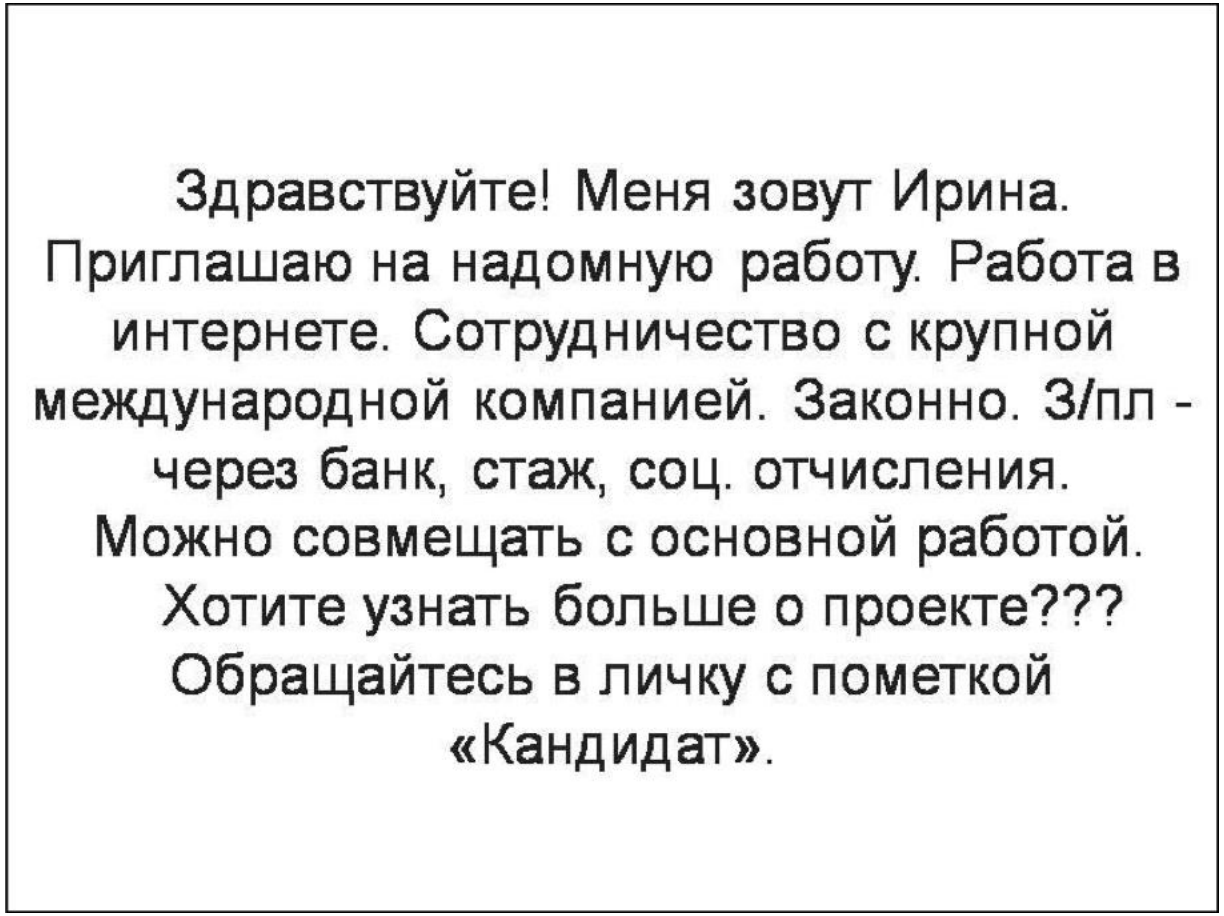
*must have for Senior ML Engineer

https://github.com/donnemartin/system-design-primer

Case study: OCR

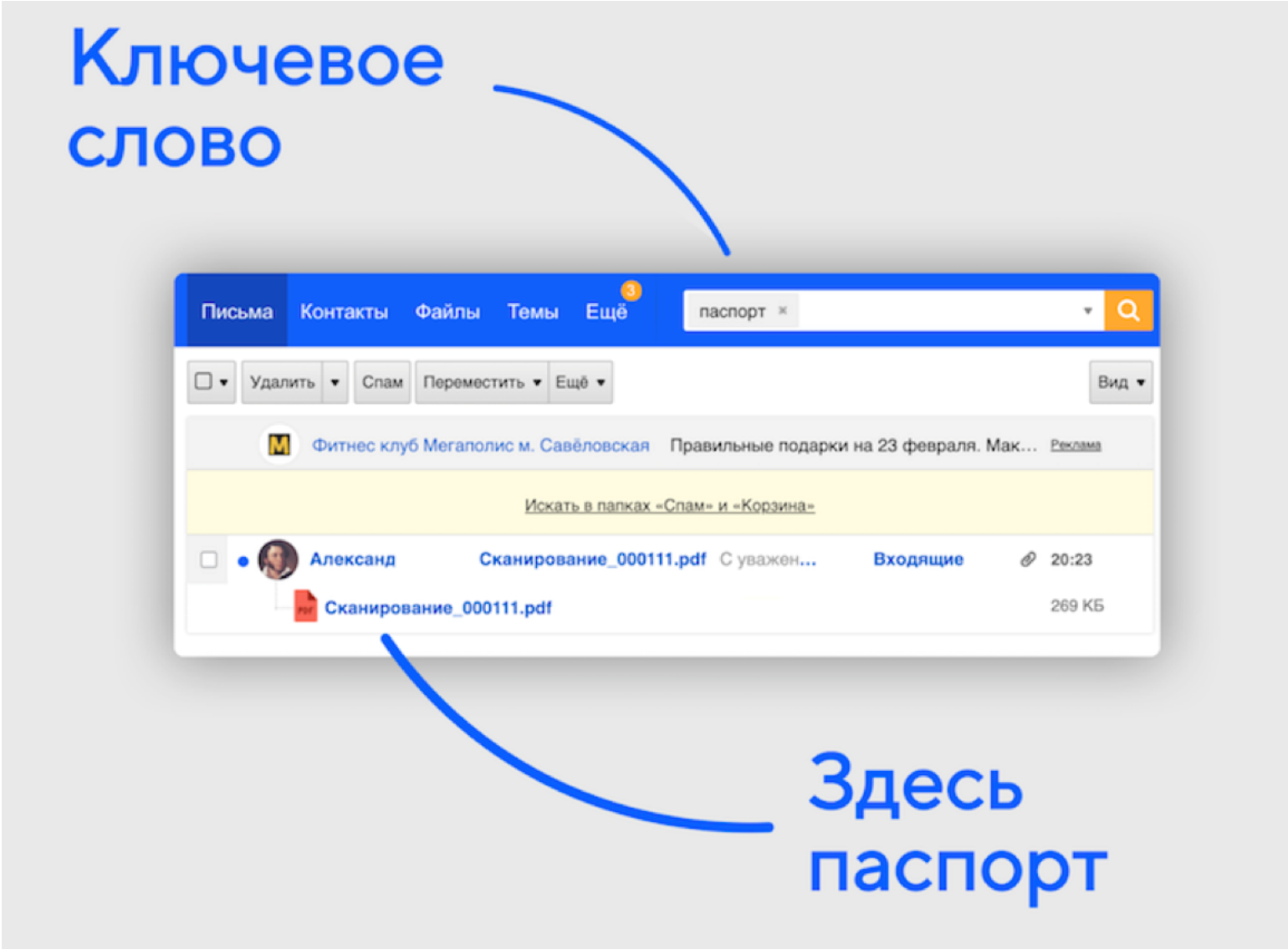
Application: Antispam

- Spam using Images/PDF
- Adaptation/Adversarial



Application: Email service

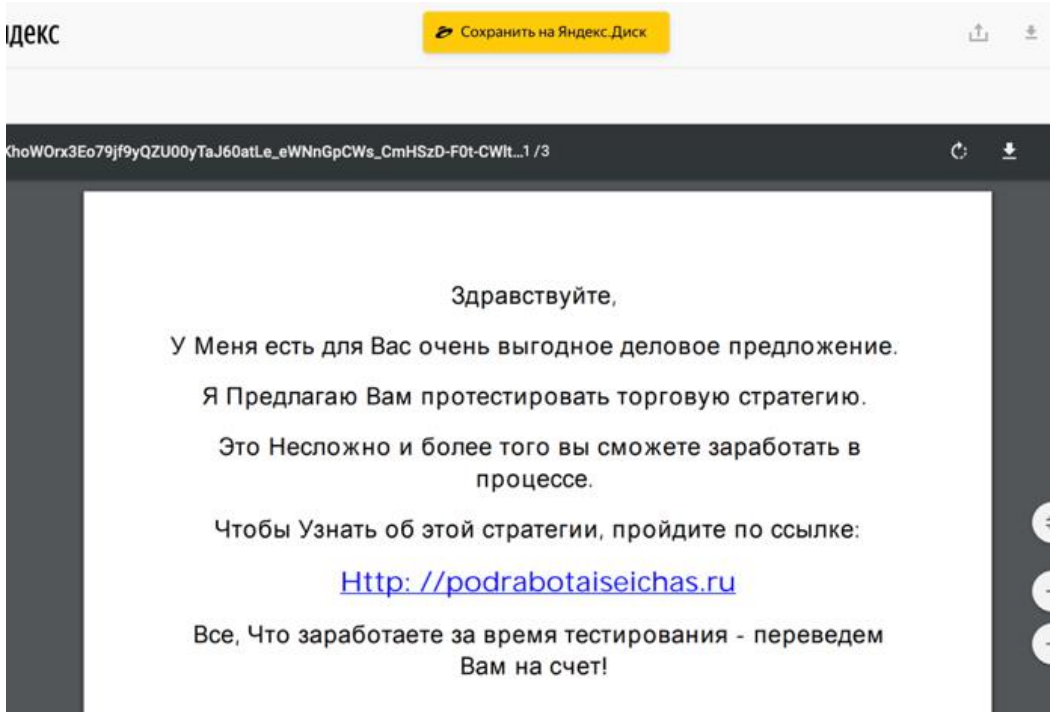
- Text search
- Document types

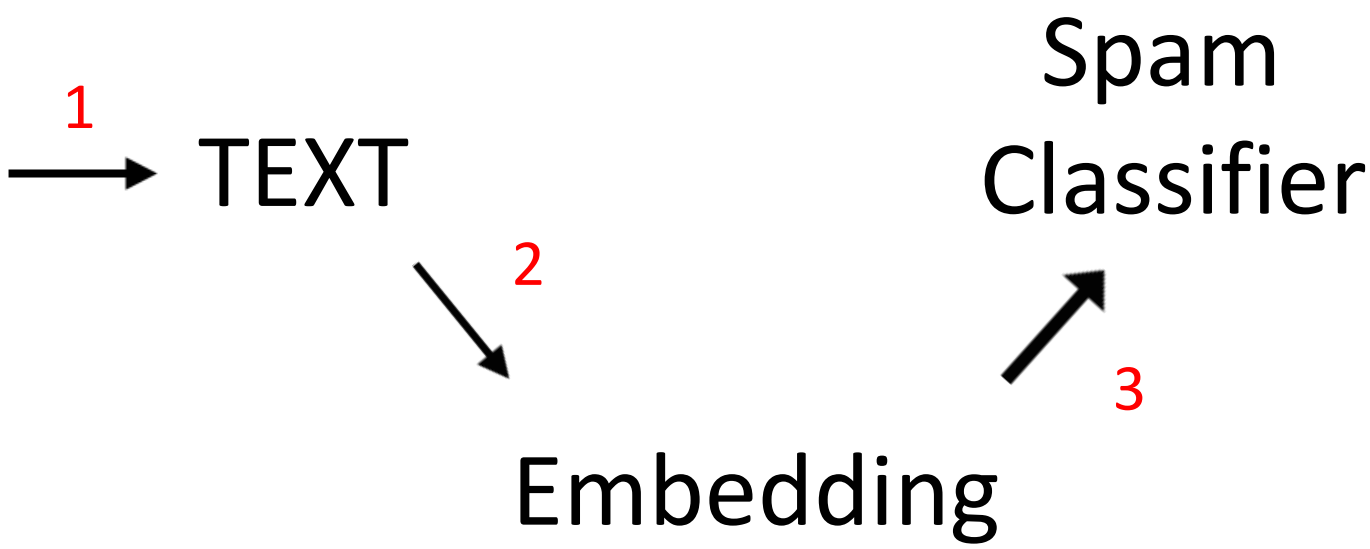


1. Problem statement

Objective

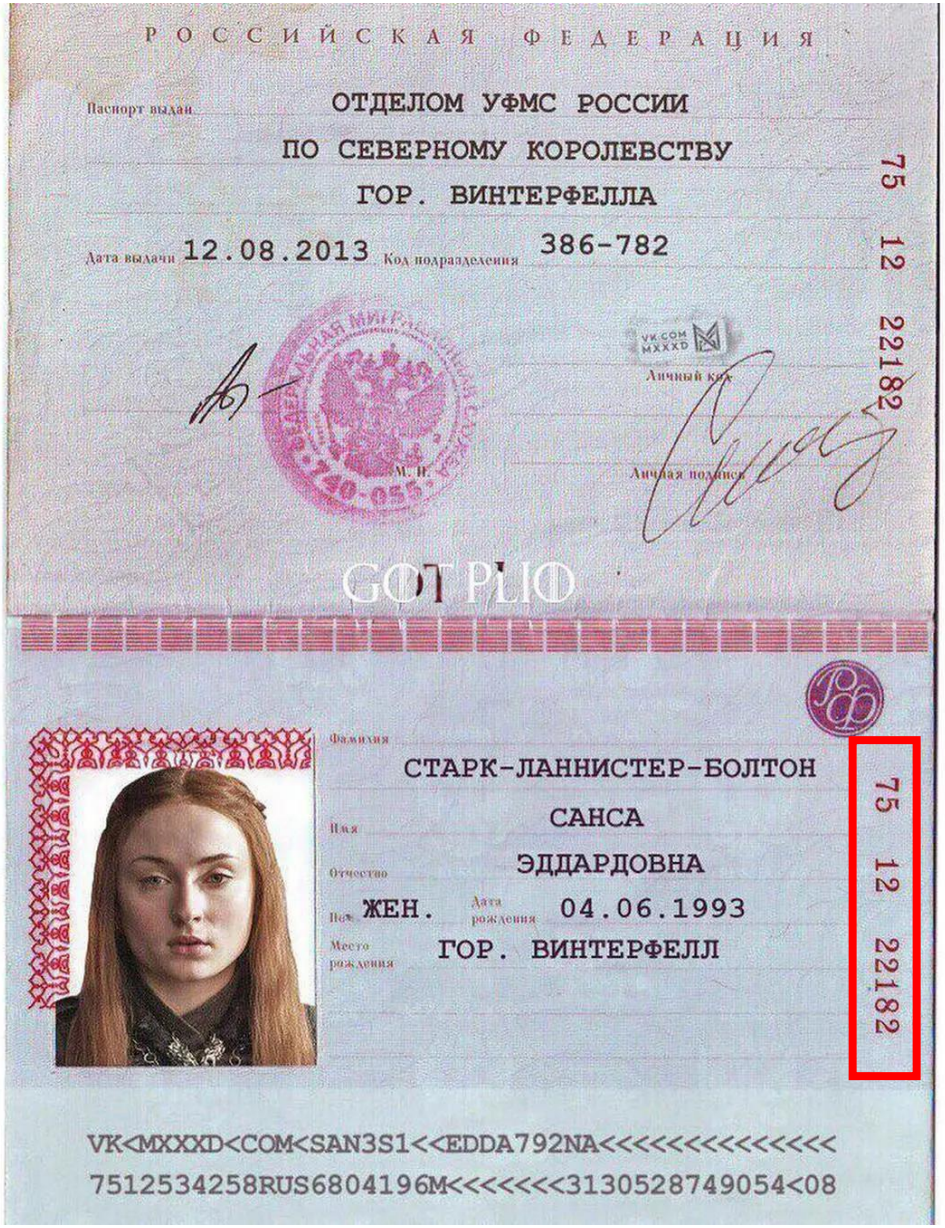
– Embedded component: for other ML or search engine





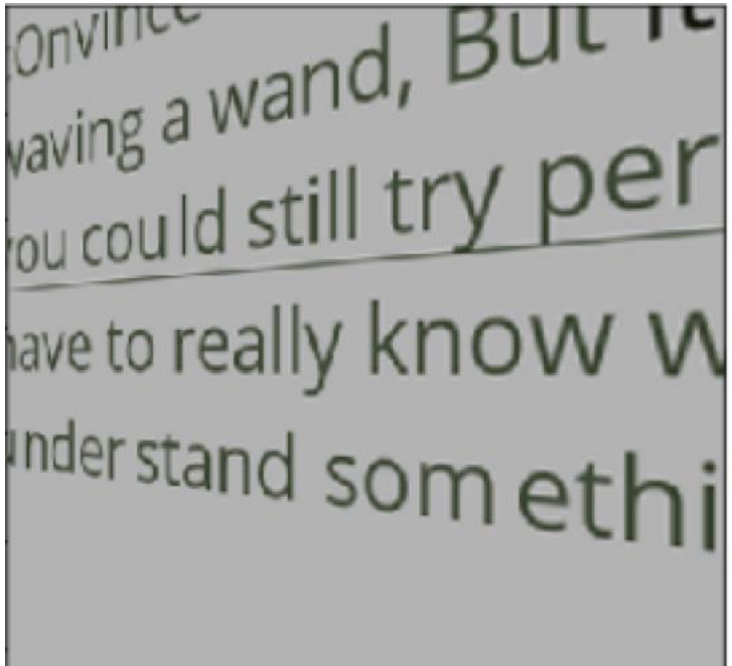
Objective

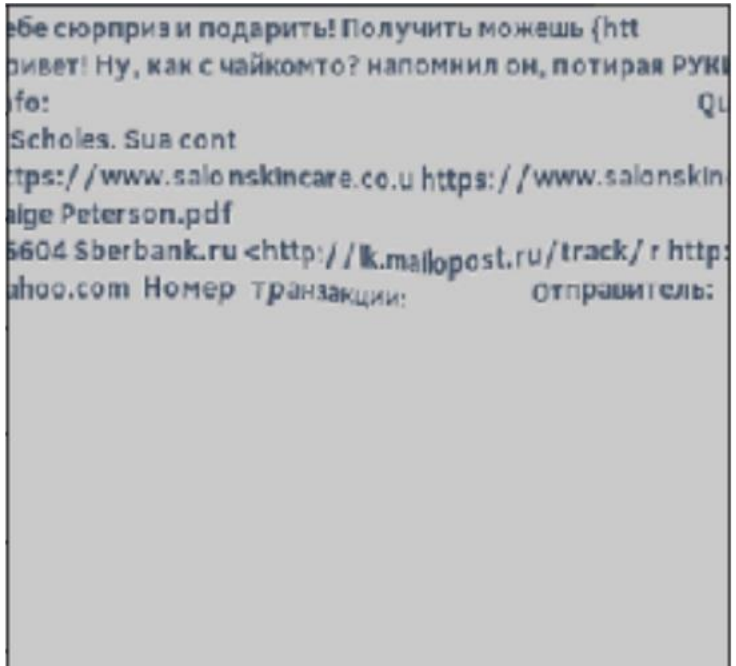
- Errors are not critical
  - != passport recognition
  - which type of error more important ?

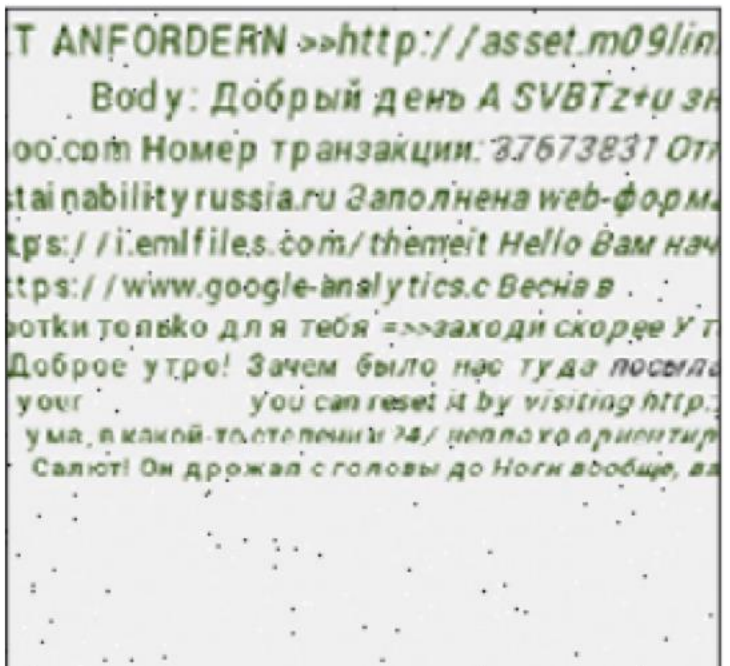


Objective

- Multiple languages
- Decent accuracy: Word Error Rate (>98%)
- **Adaptability**







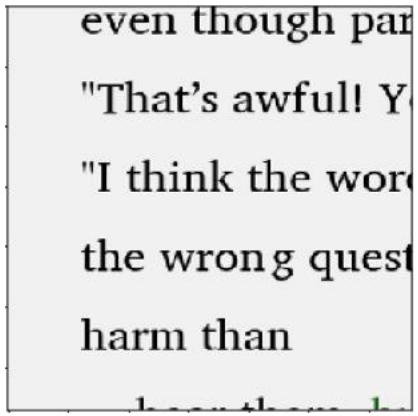
Scale & Inference budget

- 50,000 rpm
- <100ms on CPU+GPU for Antispam
- <500ms for Email/other applications

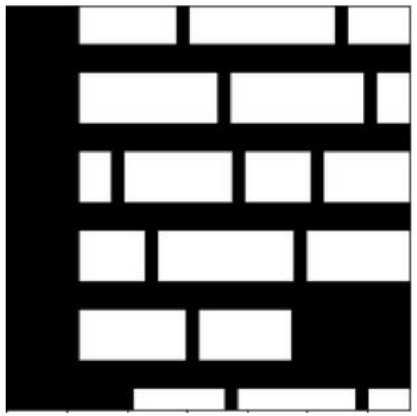
2. Data

Sources, Supervision

- Labelling is very hard
  - Doable: hard examples for recognition

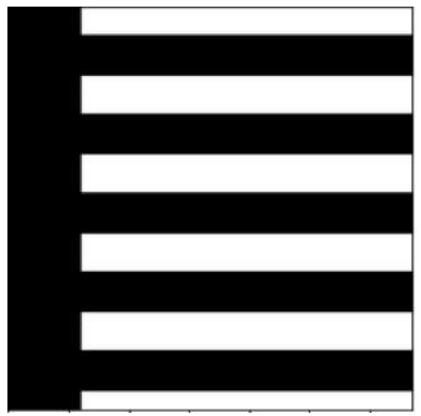


image



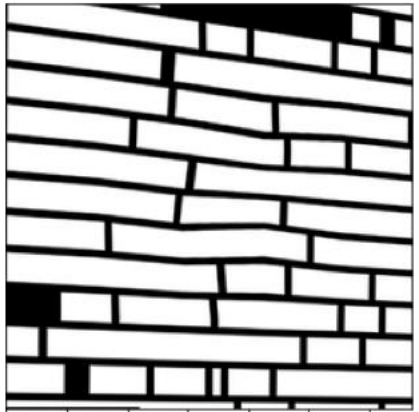
words

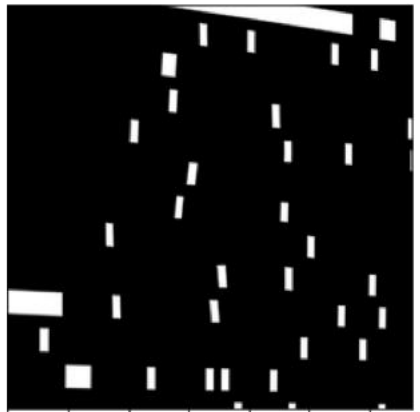
spaces

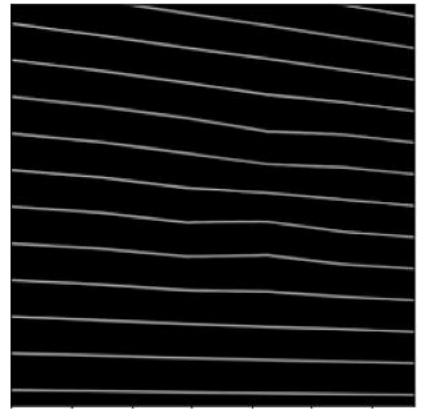


rows

delimiters

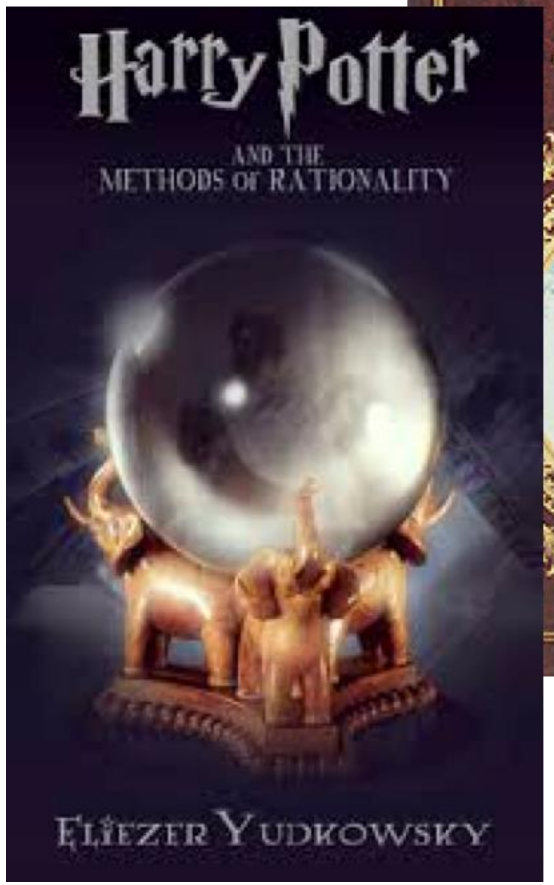






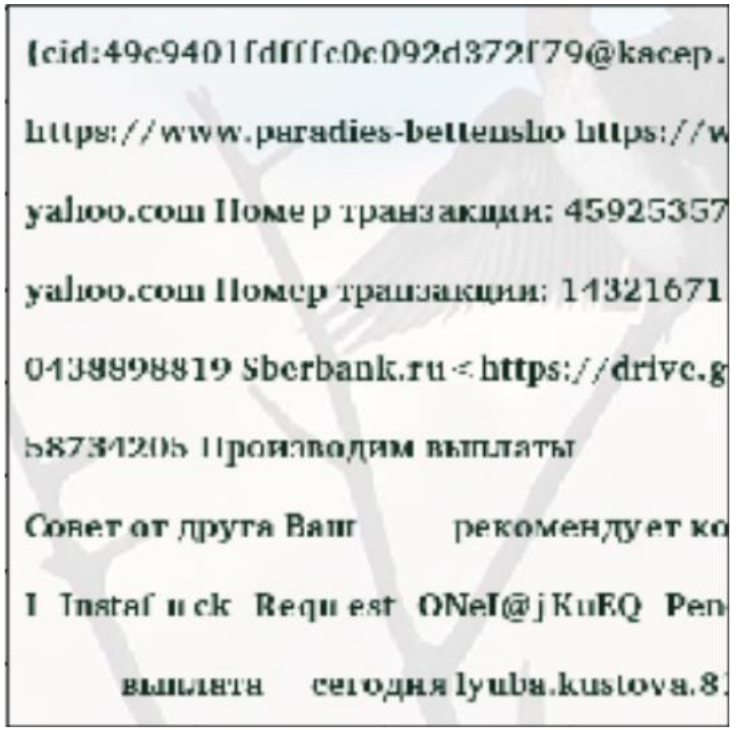
Data: generator

- Opensource Backgrounds
- Texts: literature, spam, documents

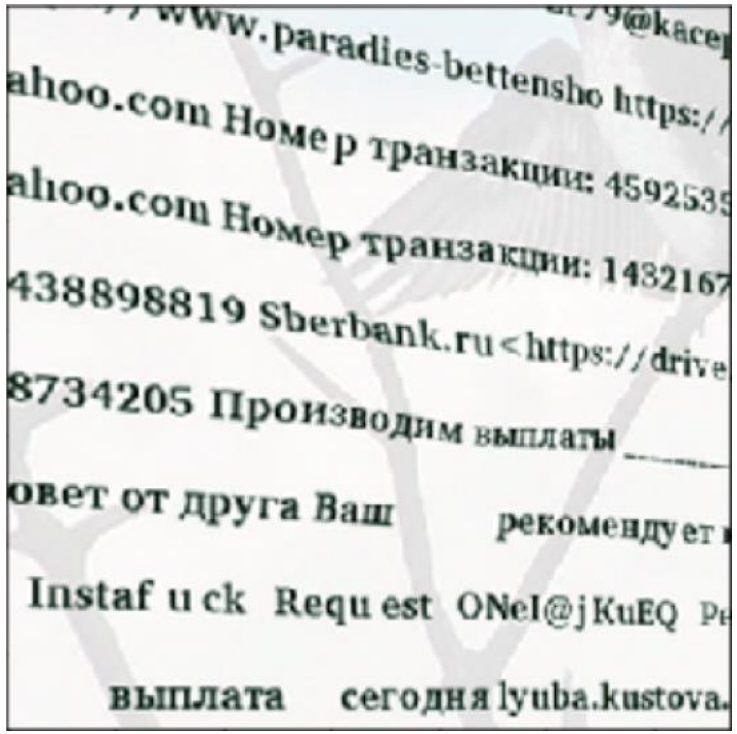


Data generator: example

Pick a background



+Text



Augment

Data: Feedback loop

- No direct feedback ;(
- Adaptation through hard examples
  - how to **mine** them? (see metrics section)

3. MVP

MVP

Heuristics, opensource?

- Tesseract doesn’t fulfill neither requirement
  - Inference: CPU, 1-3s
  - Adaptability: bad

https://github.com/tesseract-ocr/

MVP

The only option: deploy first prototype CNN

What can we learn?

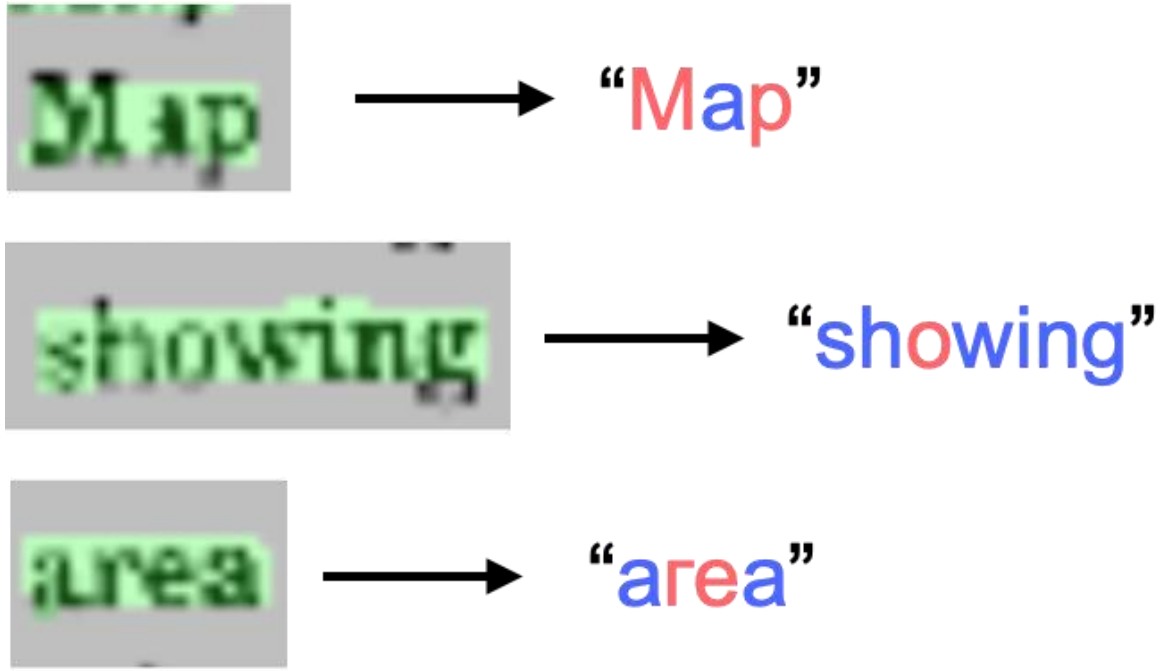
- Risks: inference budget
- Data: longtail, edge cases

New information: data & backend

- Not by images alone – PDF
  - –> one more backend + near-online
- Large images (>10mb)
  - –> significant strain on CPU
  - –> to handle all images – near-online

New information: data

1. Language mismatch



New information: data

mail.ru group

2. Trash text

→ “ЭЩКЕРЕ”

New information: data

3. “misprint” errors

→ “TabId 6.”

→ “Eigure”

4. Model

Model

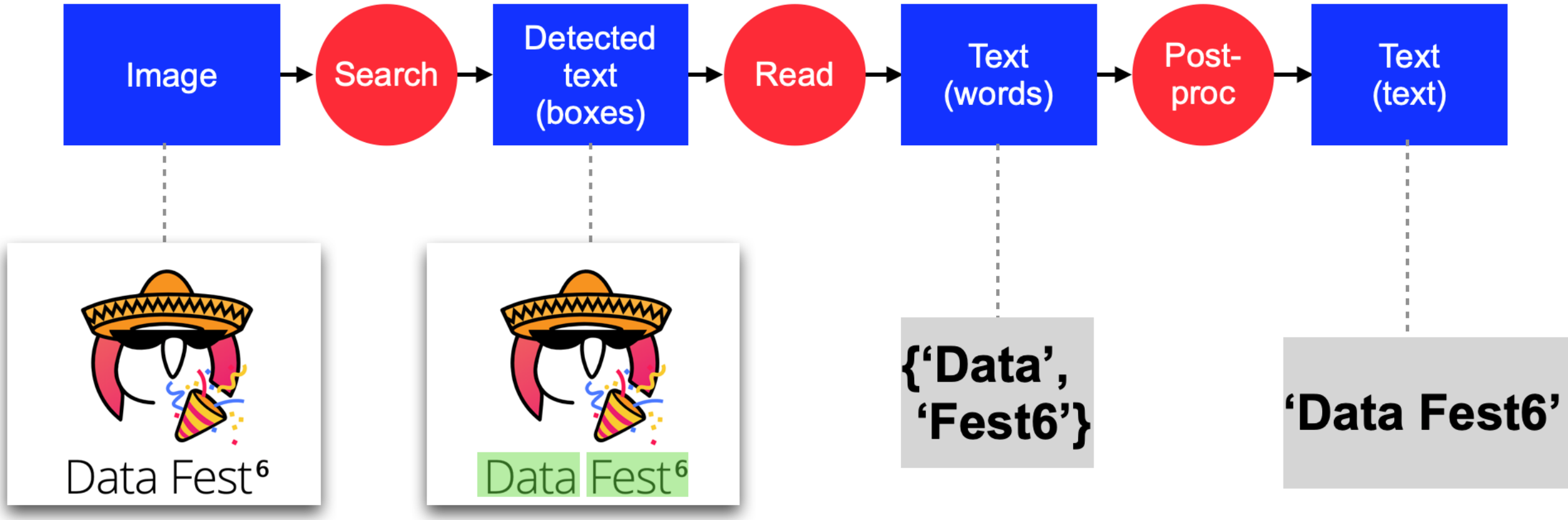


Image
Search
Detected text (boxes)
Read
Text (words)
Post-proc
Text (text)
Data Fest⁶
Data Fest⁶
{‘Data’, ‘Fest6’}
‘Data Fest6’

How to handle edge cases?

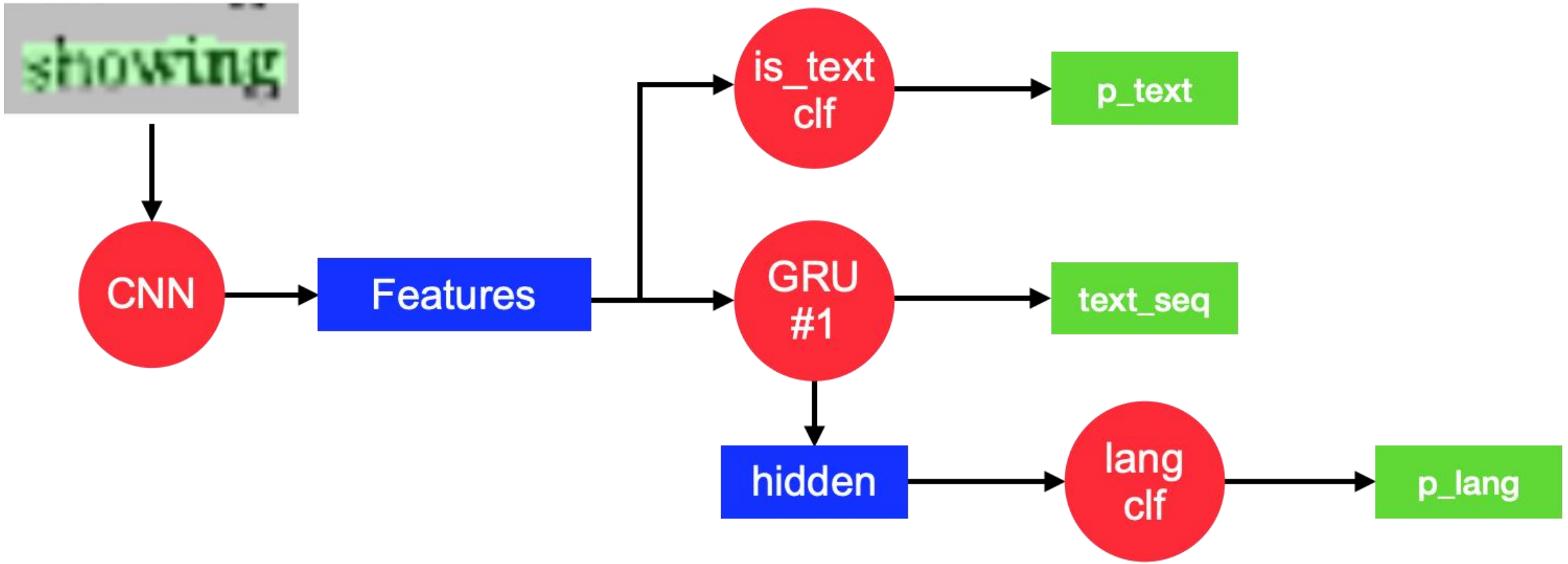
mail.ru group

1. Language mismatch
2. Trash text
3. “Misprint” errors

Edge cases

- Simple heuristics > ML
- ML > complex heuristics

Multi-head for Recognition



Misprints

How to tackle:

- Word model
- One more head
  - word, BPE, n-grams, …

→ “Tabld 6.”

→ “Eigure”

5. Metrics

5. Metrics

1. Complaints on images (antispam)
2. Added value to Classifier’s metrics
   - Average BPE length
3. Accuracy

BPE metric usage

We can use BPE to estimate trash text probability

“Hello, Made students” – High BPE length

“dsakljds saknx2 sdak99” – Low

Effective for error mining

https://www.derczynski.com/papers/archive/BPE_Gage.pdf

6. Deployment

A/B testing

- How to split ?
  - A/A test
- Which metrics ?
  - Only responsive (added value, BPE)

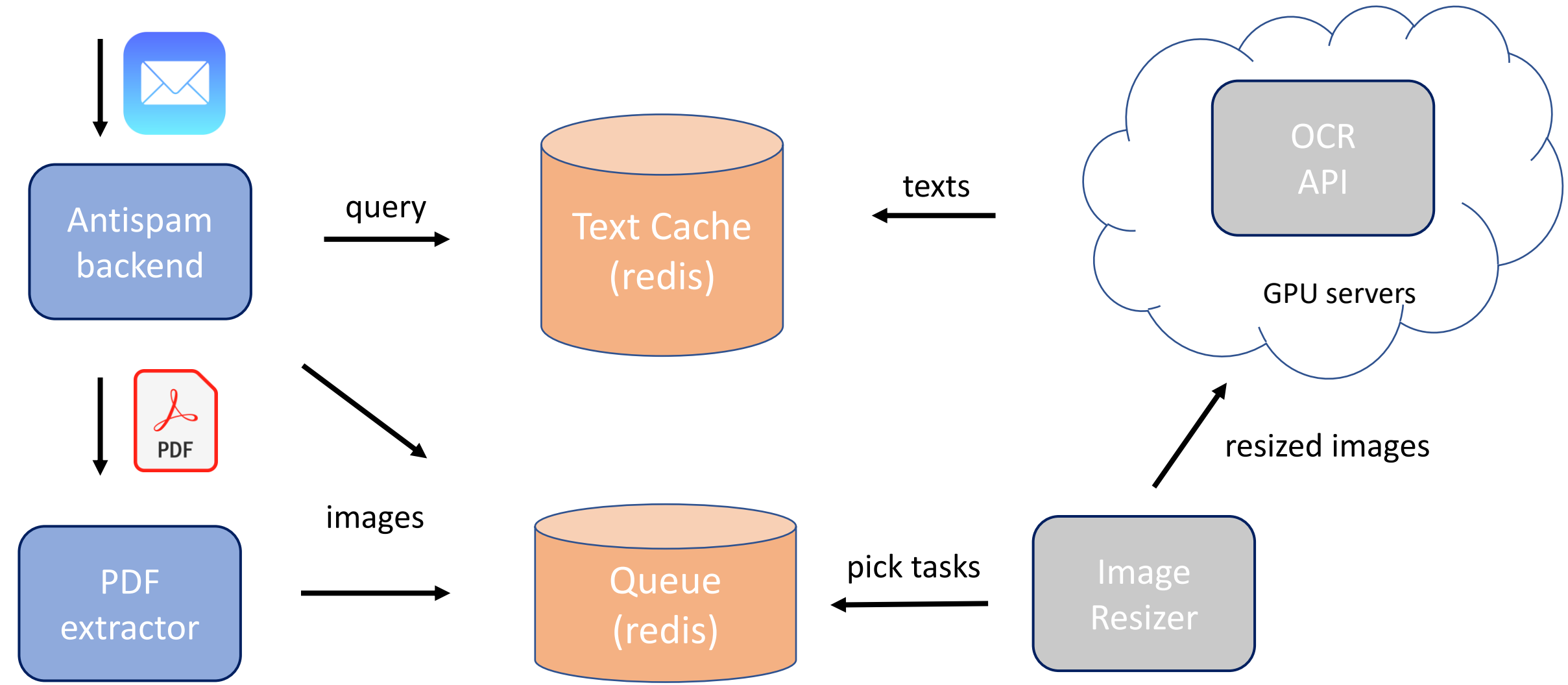
Monitoring

Find error in a pipeline as soon as possible

- Integration tests
- Data tests
- Monitor metrics
  - Technical: volume, response time, …
  - Product

System design

Antispam design: near-online



Emails with first/unique images/PDFs are post-proccessed

Case study: Face recognition on video

FR on video

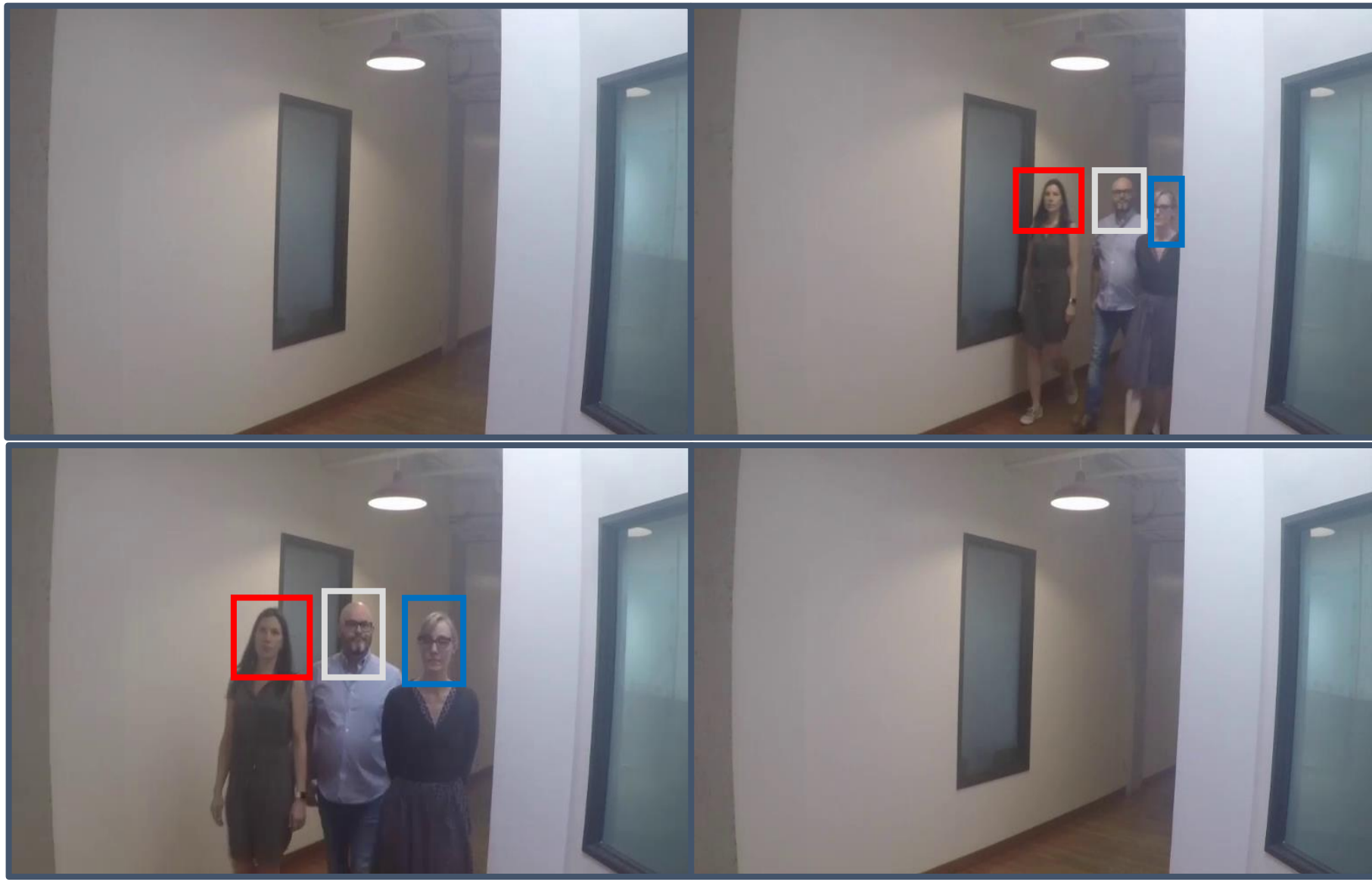
Imagine we've already have FR on photo

(see "Metric learning" lecture)

Video **>** independent sequence of photos

Application on site

- Access control (known people)
- Track intruders (unknown)



1. Problem statement

Objective

FP vs FN ?

- Access control –> very low FPR < 0.01%
- Recall (rank1) > 95%
  - How to initialize the system ?

Restrictions

- Scale: N cameras
- Inference budget: < 1s
  - people don’t like to wait

2. Data

Data Sources

Specific hardware (cameras)

–> Tuning on ”home” installation

No feedback loop ;(

3. MVP

MVP

Video = sequence of photos ;)

What can we learn?

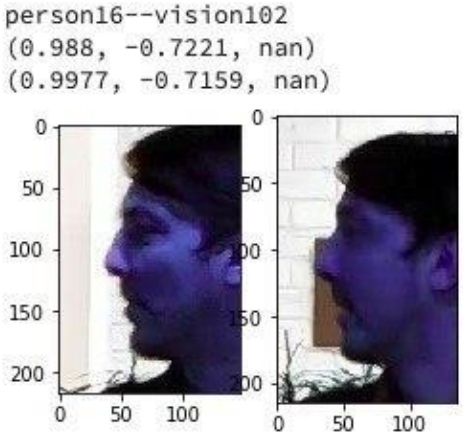
- Cameras: blurry, bad angles, too far, low resolution
- How many extra persons ?

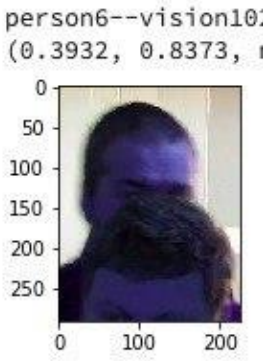
New info: blur

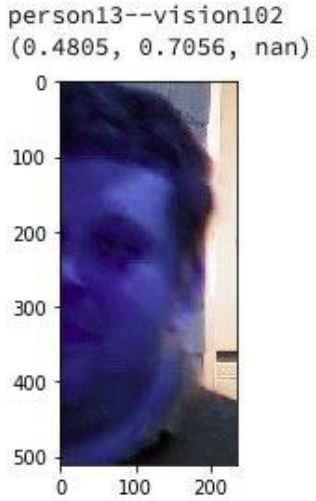
New info: Occlusion

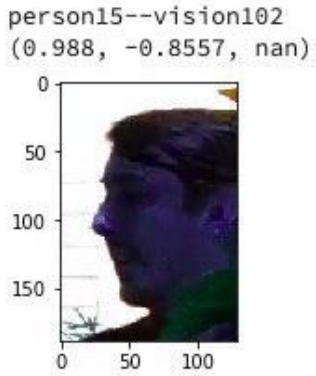
FN

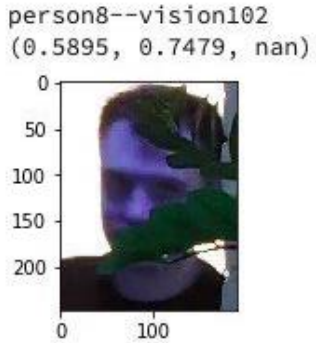
- Unbelievable amount of
  - Edge cases
  - “Splits”

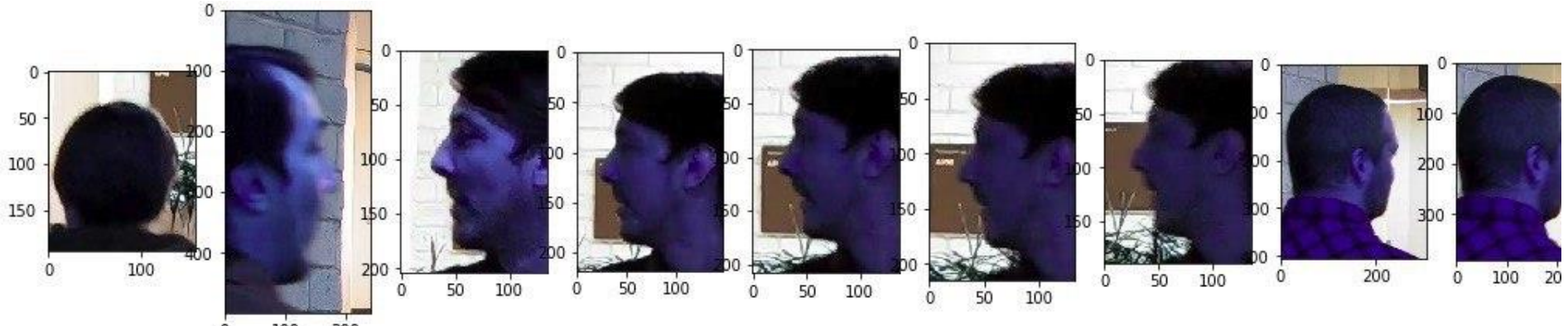












Cameras

- Bad tuning
- Weird angles

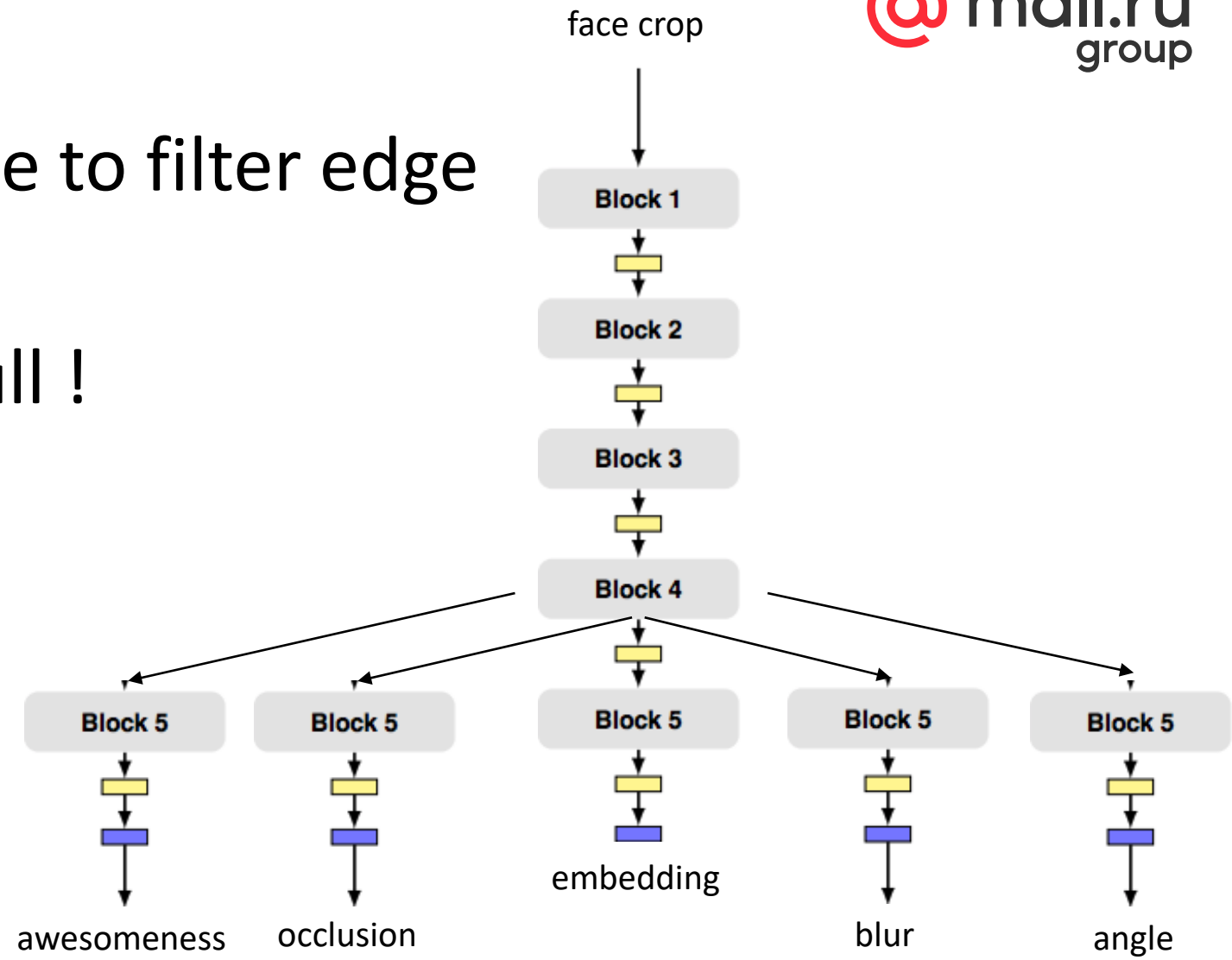
–> tests & guides

4. Model

Filtering

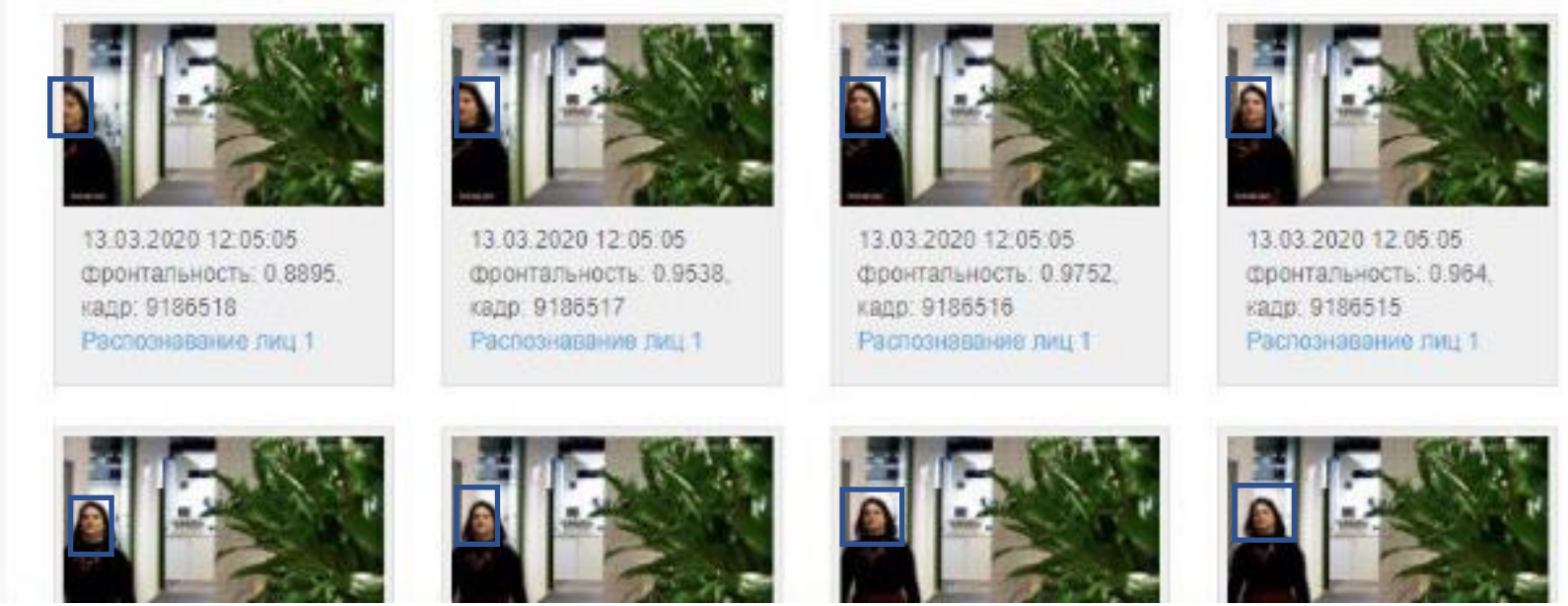
Predict every aspect of face to filter edge cases

Gives us many levers to pull !



Tracking

- Use sequence!
- Track instead of face



Tracking: example

Tracking

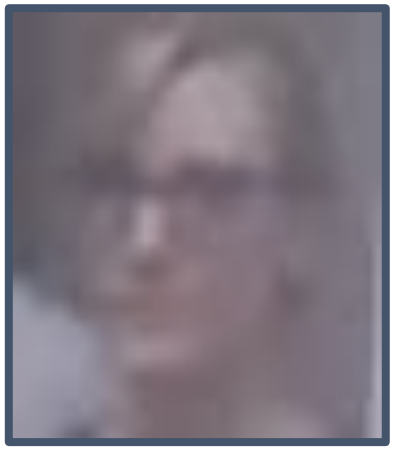
Sort/DeepSort uses:

- Bounding boxes
- Embeddings

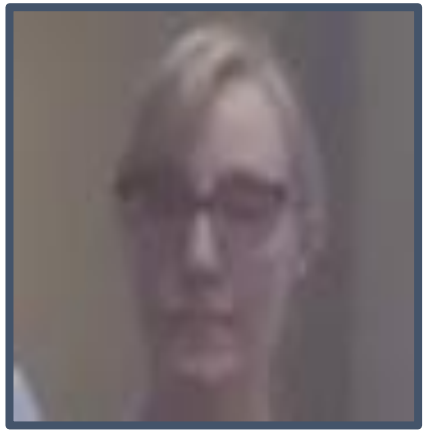
https://arxiv.org/pdf/1703.07402.pdf

Tracking: best shot

Score = Frontality + Awesomeness + Blurriness + …

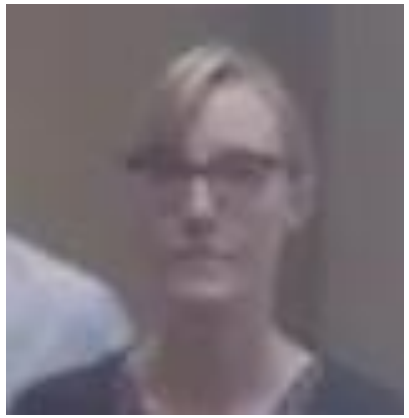


Score: 0.1



Score: 0.7

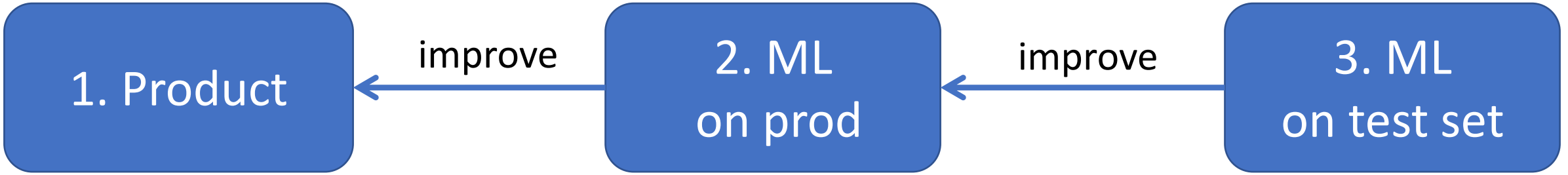
Best



Score: 0.4

5. Metrics

5. Metrics

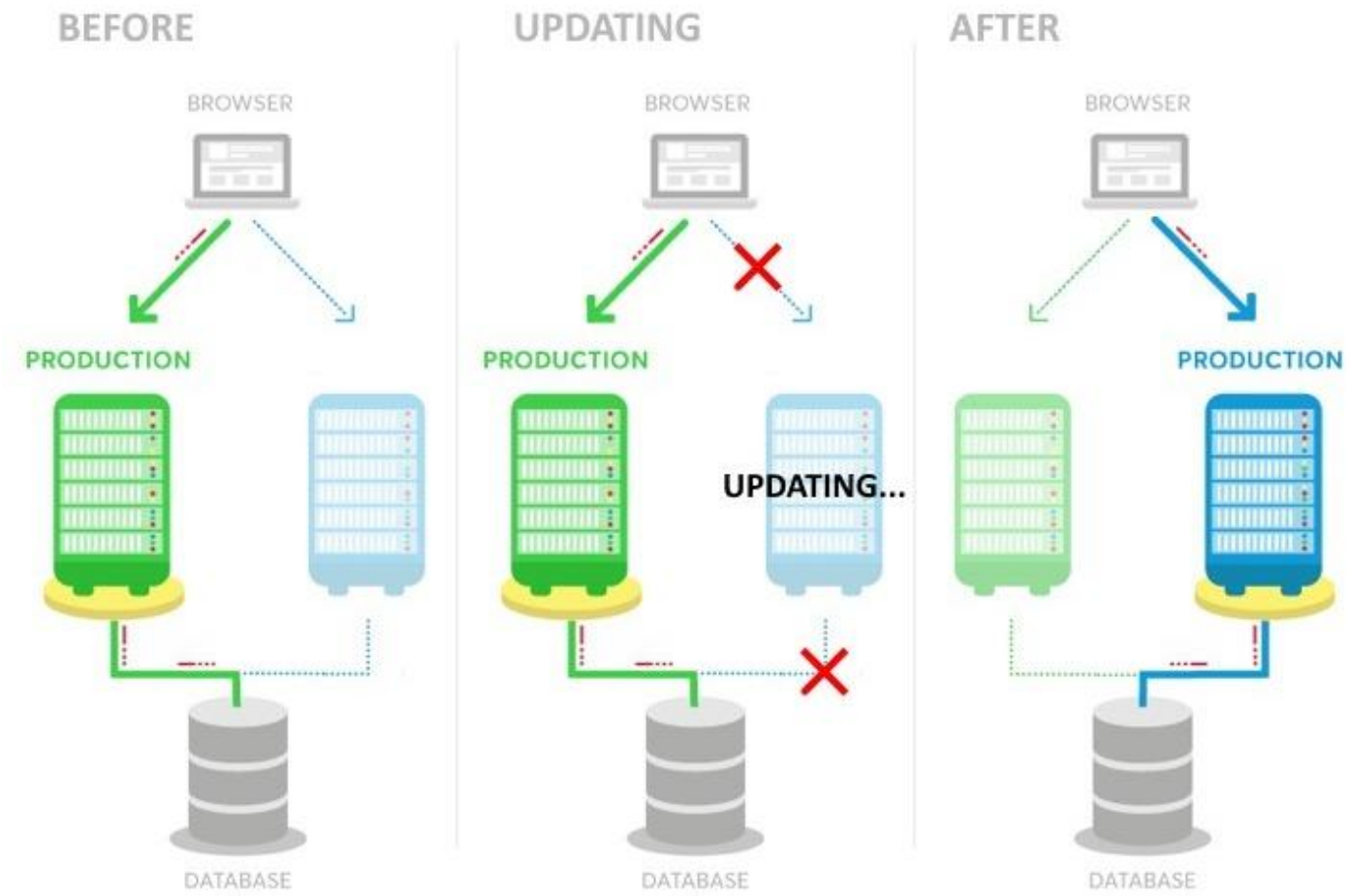


1. FPR, Rank1 (manual)
2. Feedback from users
3. Rank1

6. Deployment

A/B testing

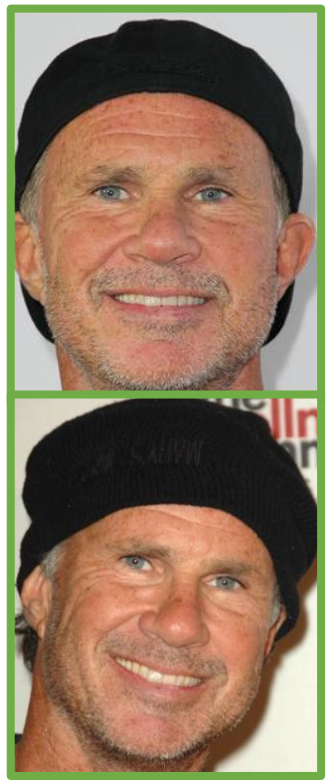
- How we A/B test with new embeddings ?
  1. Blue/Green deployment



A/B testing

- How we A/B test with new embeddings ?
  1. Blue/Green deployment
  2. **Harmonic** embeddings

Triplet loss: harmonic



Positive

minimize

Anchor

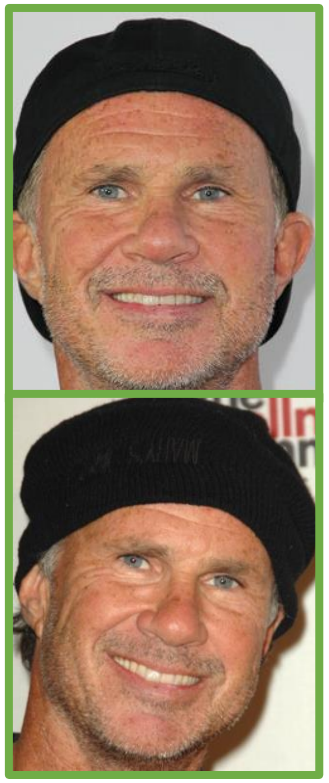
maximize

Negative

positive + α < negative

https://arxiv.org/abs/1503.03832

Triplet loss: harmonic



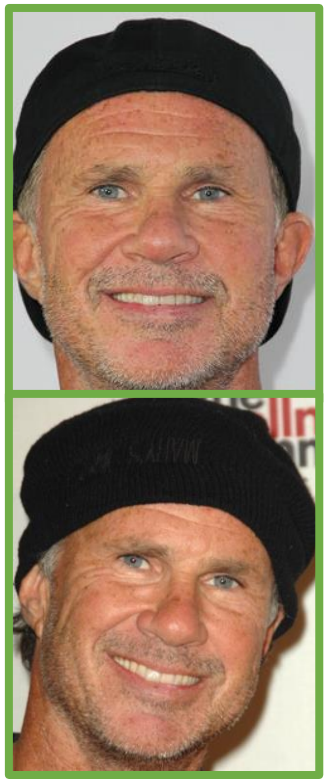
minimize

Anchor

maximize

New

Triplet loss: harmonic



Emb v2

Emb v1

minimize

Anchor

maximize

Emb v2

Emb v2

Emb v1

Mix embedding versions during training

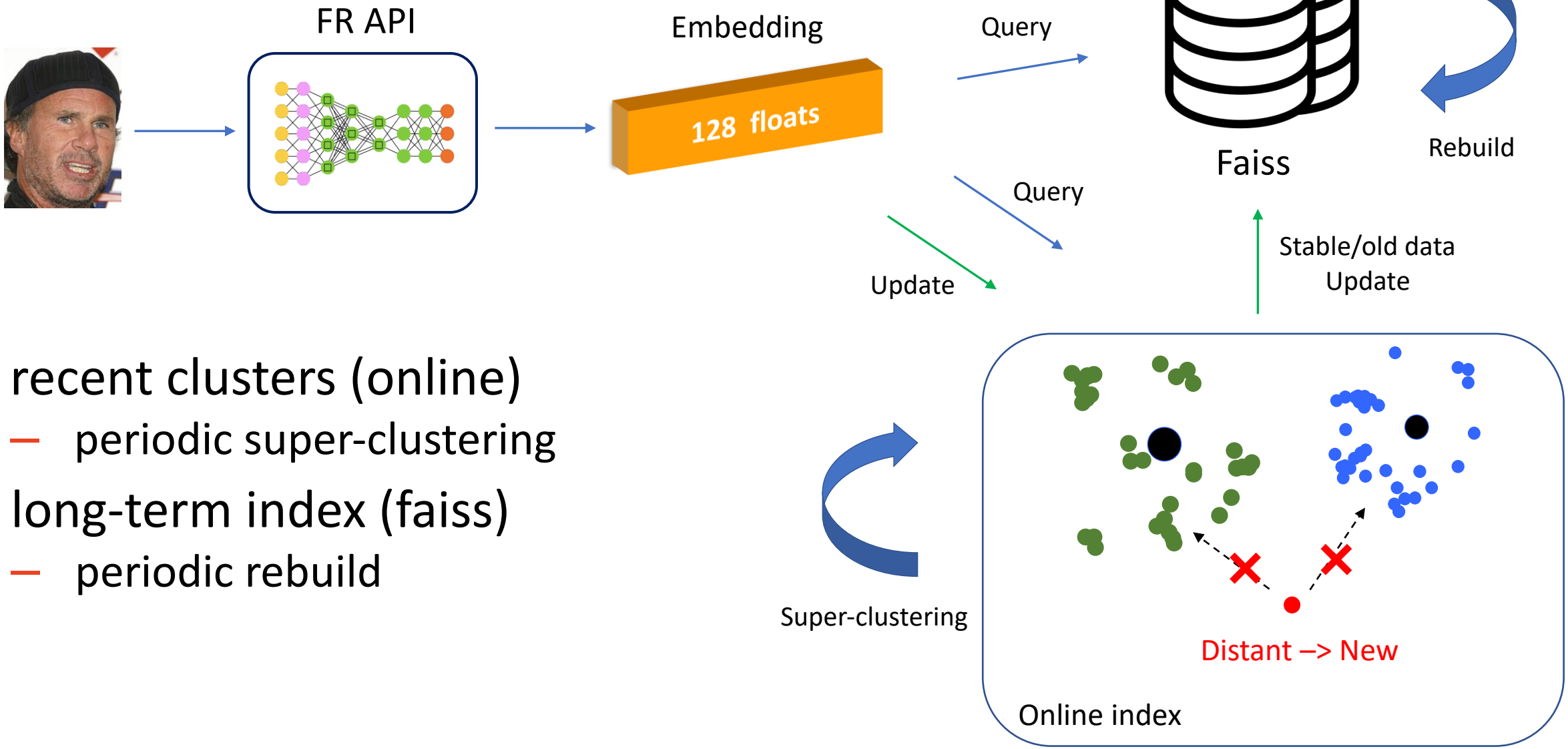
System design

Index

- KNN Index
  - not easy to update it on the fly
  - –> Store only stable data
- Online clustering for new

Index (FAISS/Custom)

Index



- recent clusters (online)
  - periodic super-clustering
- long-term index (faiss)
  - periodic rebuild

System design

1

API

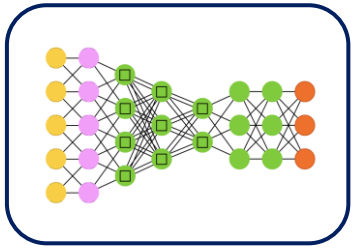
2

Decoding/
Tracking

3, 5

emb. + id?

FR API



Search

4

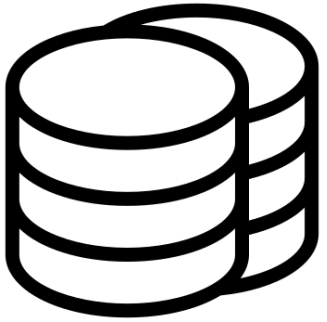
6

Track: best shot

Clustering

7

Emb.
update



Index (FAISS/Custom)

Post-processing/Clustering

8 Result: person_id

Takeaways

Takeaways

1. Problem statement: Objective, Inference budget
2. Data: Sources, Signal, Feedback Loop
3. MVP
4. Model: Features, Arch, Loss, …
5. Evaluation metrics (test –> prod-ml –> product)
6. Deployment: A/B testing, Monitoring
7. System design: Scale, Trade-offs

Контакты

tg: @ed_tyantov

tyantov@corp.mail.ru

академия больших данных

mail.ru group

made