Bringing automation and fairness to identity verification on the internet with deep learning

Olivier Koch, VP of AI - Onfido
Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics
Special credits to Yuanwei Li, Martins Bruveris, and Richard Tomsett
Who are we?

Onfido is an online identity verification company.

We let our customers verify the identity of their users.
Current industries

- Financial Services
- Cryptocurrencies
- Marketplaces
- E-commerce
- Transportation
- Gaming
- Healthcare

The future

- Hotels
- Airlines
- Telecoms
- Government
Onfido’s 3 layers of identity verification

1. Do you have a **genuine** ID?
2. Are you a **real life** human?
3. Does your face **match** your ID?
Document Verification

+ Thousands of document types
+ Constantly changing attack vectors
+ Variable image quality (API vs SDK)
+ Very low signal-to-noise ratio
Biometric Verification

- Low friction and accessibility requirements
- Bias reduction
- Deepfakes and injection attacks
Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics
Automation brings several key benefits:

- Remove human variance
- More $ efficiency
- Better privacy
- Speed

At the cost of:

- More complexity (ML)
- AI risks (bias)
Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics
Let’s focus on document verification
Machine learning problem statement

Determine whether a document is fraudulent or not, given a large dataset of genuine samples and a (smaller) dataset of frauds

Across thousands of document types

Binary classification problem across many categories
Key metrics

**FAR:** False Acceptance Rate = \[
\frac{\# \text{ accepted}}{\# \text{ submitted}} \quad (\text{all frauds})
\]

**FRR:** False Rejection Rate = \[
\frac{\# \text{ rejected}}{\# \text{ submitted}} \quad (\text{all genuine})
\]
Supervision beats unsupervised by a wide margin

Unsupervised (GMM): 5% FRR, 50% FAR
Supervised (LR, auto-encoders): 5% FRR, 10% FAR
Three approaches

1. Train a single model for all doc types

1. Train a model per doc type

1. Meta-learning
Model-Agnostic Meta-Learning (Finn, et al. 2017)

Source: Meta Learning, learning to learn fast, Lilian Weng
Model-Agnostic Meta-Learning (Finn, et al. 2017)

Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(T)$: distribution over tasks
Require: $\alpha, \beta$: step size hyperparameters

1: randomly initialize $\theta$
2: while not done do
3: Sample batch of tasks $T_i \sim p(T)$
4: for all $T_i$ do
5: Evaluate $\nabla_\theta \mathcal{L}_{T_i}(f_\theta)$ with respect to $K$ examples
6: Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta)$
7: end for Note: the meta-update is using different set of data.
8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta_i'})$
9: end while

The general form of MAML algorithm. (Image source: original paper)
One model per document type

- Italian ID 2002
- French passport 2022
- California DL 2018

Training -> Model
Meta-learning

Italian ID 2002

French passport 2022

Meta-learner
Meta-learning

Italian ID 2002

French passport 2022

California DL 2018

Meta-learner

Better model
Particularly valuable for long-tail distributions
Meta-training

Italian ID 2002

French passport 2022

Query set
Outer training loop

Support set
Inner training loop

Each composed of K genuine and K fraud samples

Meta-learner
Meta-validation: train on support, evaluate on query

Unseen data

Query set  Support set
Outer training loop  Inner training loop

California DL 2018 → Better model
Validation split

- Split A: all data in training
- Split B: all top docs in validation
- Split C: a few top docs in validation

We present results on Split B
## Experimental setup

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML 1</td>
<td>MAML with outer_lr=0.0001, inner_lr=0.1</td>
</tr>
<tr>
<td>MAML 2</td>
<td>MAML with outer_lr=0.0001, inner_lr=2.0</td>
</tr>
<tr>
<td>Pretrain</td>
<td>Supervised pre-training using MAML without inner loop. outer_lr=0.0001</td>
</tr>
<tr>
<td>Baseline</td>
<td>Random weight initialisation</td>
</tr>
</tbody>
</table>

We use the code from the original paper:

https://github.com/cbfinn/maml
<table>
<thead>
<tr>
<th>Fine-tuning method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No fine-tuning (zero-shot inference)</td>
<td>The model weights from the training experiments are used directly for zero-shot inference without any fine-tuning on doc-specific training samples.</td>
</tr>
<tr>
<td>Fine-tune by steps</td>
<td>The model weights are fine-tuned on doc-specific training samples. <em>We only use 1 genuine and 1 fraud samples for training.</em> The performance is evaluated after a few steps (1, 2, 3, 4, 5, 10) of model updates on the same pair of training examples.</td>
</tr>
<tr>
<td>Fine-tune by epochs</td>
<td>The model weights are fine-tuned on doc-specific training samples. <em>We use a lot of genuines (thousands) and varying number of frauds for training.</em> Fine-tuning is conducted for 60 epochs of the genuine data. Performance is evaluated when different number of training frauds are used.</td>
</tr>
<tr>
<td>Fine-tuning settings</td>
<td>FAR@FRR=0.02</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Baseline (random weights)</td>
</tr>
<tr>
<td>No fine-tuning (zero-shot inference)</td>
<td>0.9536</td>
</tr>
<tr>
<td>Fine-tune by steps</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9536</td>
</tr>
<tr>
<td>1</td>
<td>0.9536</td>
</tr>
<tr>
<td>1</td>
<td>0.9536</td>
</tr>
<tr>
<td>1</td>
<td>0.9536</td>
</tr>
<tr>
<td>1</td>
<td>0.9536</td>
</tr>
<tr>
<td>1</td>
<td>0.9536</td>
</tr>
<tr>
<td>Fine-tune by epochs</td>
<td></td>
</tr>
<tr>
<td>All 0</td>
<td>0.6337</td>
</tr>
<tr>
<td>All 1</td>
<td>0.5776</td>
</tr>
<tr>
<td>All 5</td>
<td>0.4587</td>
</tr>
<tr>
<td>All 10</td>
<td>0.3948</td>
</tr>
<tr>
<td>All 50</td>
<td>0.2362</td>
</tr>
<tr>
<td>All 100</td>
<td>0.2028</td>
</tr>
<tr>
<td>All 500</td>
<td>0.2019</td>
</tr>
<tr>
<td>All 1000</td>
<td>0.1576</td>
</tr>
</tbody>
</table>
MAML outperforms the best pretraining baseline on the zero-shot task (albeit by a small margin): 0.364 < 0.382
Results on validation set with doc split B (21 docs held out)

At the low fraud data regime, MAML outperforms pretraining significantly.
At the high fraud data regime, all methods are on par.
Results on validation set with doc split B (21 docs held out)

<table>
<thead>
<tr>
<th>Fine-tuning settings</th>
<th>FAR@FRR=0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAML 1</td>
</tr>
<tr>
<td>No fine-tuning (zero-shot inference)</td>
<td>0.3646</td>
</tr>
<tr>
<td>Fine-tune by steps</td>
<td></td>
</tr>
<tr>
<td>1 step</td>
<td>0.3596</td>
</tr>
<tr>
<td>2 steps</td>
<td>0.3596</td>
</tr>
<tr>
<td>3 steps</td>
<td>0.3591</td>
</tr>
<tr>
<td>4 steps</td>
<td>0.3591</td>
</tr>
<tr>
<td>5 steps</td>
<td>0.3591</td>
</tr>
<tr>
<td>10 steps</td>
<td>0.3591</td>
</tr>
<tr>
<td>Fine-tune by epochs</td>
<td></td>
</tr>
<tr>
<td>All 60 epochs</td>
<td>0.3591</td>
</tr>
<tr>
<td>All 1 epoch</td>
<td>0.3591</td>
</tr>
</tbody>
</table>

Fine-tuning on a single sample, the best performance is reached with the same number of training steps that was used during training (1 step).
MAML1: inner loop learning rate too small (lr = 0.1)

Zooming in on the outer loop training (pre-update loss, post-update loss)

MAML2: inner loop is working (lr = 2.0)
Our results support a “feature reuse” scenario

Figure 1: **Rapid learning and feature reuse paradigms.** In Rapid Learning, outer loop training leads to a parameter setting that is well-conditioned for fast learning, and inner loop updates result in significant task specialization. In Feature Reuse, the outer loop leads to parameter values corresponding to reusable features, from which the parameters do not move significantly in the inner loop.
MAML allows to get the best of both worlds:

- Best performance in low-data regime
- On-par with pretraining in high-data regime
Outline

1. Who are we?
2. Why automate identity verification?
3. Meta-learning for document verification
4. Bias reduction for biometrics
Several definitions of bias

Demographic parity

Equality of opportunity

Equality of odds

Predictive parity

Source: Google glossary on fairness
Equality of opportunity

Candidates are equally likely to be admitted irrespective of which group they belong to, as long as they are qualified.

Proposed metric for fairness in identity verification

FRR should be the same across groups.

Measure FRR/group and normalize by overall FRR.

Ratio > 1: group is over-rejected

Ratio < 1: group is under-rejected
Bias mitigation: demographic differential for Motion

Source: “Building without bias”, Onfido

FRR bias against overall population (1.0 = no bias)

95% confidence intervals
Bias mitigation: demographic differential for Motion

From our latest white paper “Building without bias”

Gender bias

In regard to gender we observe some bias between male or female, with a ratio of 0.87 for male and 1.18 for female.

<table>
<thead>
<tr>
<th></th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group FRR / Overall FRR</td>
<td>0.87</td>
<td>1.18</td>
</tr>
<tr>
<td>(95% confidence interval)</td>
<td>(0.82 - 0.92)</td>
<td>(1.11 - 1.26)</td>
</tr>
</tbody>
</table>
Bias mitigation: demographic differential for Motion

From our latest white paper “Building without bias”

Age bias

In regard to age groups we see a tight grouping of ratios in all but the over 50 group.

<table>
<thead>
<tr>
<th>Group FRR / Overall FRR (95% confidence interval)</th>
<th>&lt;25</th>
<th>25-30</th>
<th>30-40</th>
<th>40-50</th>
<th>&gt;50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.89</td>
<td>0.83</td>
<td>0.87</td>
<td>1.24</td>
<td>1.71</td>
</tr>
<tr>
<td>(95% confidence interval)</td>
<td>(0.81 - 0.96)</td>
<td>(0.76 - 0.93)</td>
<td>(0.80 - 0.95)</td>
<td>(1.07 - 1.42)</td>
<td>(1.51 - 1.95)</td>
</tr>
</tbody>
</table>
Reducing bias, practical considerations

Modify the dataset

Change the training procedure

Apply post-processing to the output of the model
Conclusion

Identity verification is a core function of our digital lives.

Automating identity verification brings many benefits.

Meta-learning > supervised learning >> unsupervised

Bias matters and we propose a pragmatic approach to it.
Future areas of research

Better meta-learning models

Self-supervised learning

Generative models for realistic synthetic data