1. Supplementary Material

1.1. Implementation Details

For our industrial use cases we use an object-detection model, specifically the state-of-the-art YoLo [4], to first extract the part of the document containing the signature. In both use cases we trained an ad-hoc YoLo model to extract the signature on the specific distribution of data. In general, our trained YoLo models are capable to extract the signatures with good performances. Nevertheless, a small amount of errors from Yolo due to imperfect regions (e.g. a part of the signature is not included in the region extracted by YoLo) are included into our pipeline and can slightly affect the performances of the signature verification. It is out of the scope of this paper to evaluate the performance of the YoLo models.

For all models, for the real-world use cases we perform early stopping on the validation set, using the weights where the validation set has the higher F1-score. Conversely, for the validation on public datasets, we follow the experimental protocol described in [2] and we do not perform early stopping: we train the models for 20 epochs and take the final weights. In the following we report the implementation details for all models.

1.1.1 SignatureMatching Model

We resize each grayscale image to 32×256 pixels and normalize pixel values to $[-1.0, 1.0]$. The image embedding part is the encoder of ASTER [5] with a final bidirectional LSTM with 256 hidden units, which produces an image embedding of dimension 64×512. The encoder is initialized with the weights of the pretrained text recognition model and then fine tuned together with all other weights in the overall architecture. On top of the matrix S we first apply 16 convolutional filters with kernel size 5×5, then a max pooling operation with kernel size 5×5, and finally 32 convolutional filters with kernel size 5×5. With the configuration used in our experiments for the image embedding, we obtain a final vector of 2048 features. The final 2-layers FC module applied to the concatenation of the features is composed by a first layer producing 512 output units, followed by 0.5 dropout, then the last layer produces 2 output units. We initialize the weights of the two convolutional layers and the final two FC layers with Xavier initialization [3]. The model is trained with mini-batch gradient descend with 0.001 learning rate and 0.9 momentum and batch size equal to 128 in the cheque use case, and 0.005 learning rate with 0.9 momentum and batch size 16 in the contracts use case and for public datasets. Every 20 steps the learning rate is decreased by a factor $\gamma = 0.1$.

1.1.2 Siamese Model

For the siamese model we use as image encoder the ResNet18 model pretrained on the ImageNet dataset [1]. To do this, we use the weights provided by PyTorch[1] 1.6.0. The final FC layer is replaced by a first FC layer with 1024 units and a second FC layer with 128 units, and a dropout with probability 0.5 between them. The weights of these two FC layers are initialized with Xavier initialization [3]. The input images are resized to 224×224 dimension and rescaled to $[0, 1]$, as originally done for the ImageNet dataset. The model is trained with Adam optimizer, with learning rate 10^{-4} using fuzzy factor equal to 10^{-8} and weight decay equal to 0.0005. The batch size is 128 in the cheque use case and 16 in the contracts use case and for public datasets. Every 20 steps the learning rate is decreased by $\gamma = 0.1$.

The model is trained with a contrastive loss fed with the cosine distance between the two vector representations:

$$S_s = \cos(emb(I_1), emb(I_2)) \tag{1}$$

where $emb(I_1)$ and $emb(I_2)$ are the feature vectors computed by the siamese model on the input images I_1 and I_2 respectively. The contrastive loss used is:

$$L = \alpha l \left(1 - S_s\right)^2 + (1 - l) \max\{m - \left(1 - S_s\right), 0\}^2, \tag{2}$$

where *m* is the margin and α balances between genuine and forgery pairs.

1.1.3 SignatureMatchingR Model

We use as image encoder the ResNet18 model pretrained on the ImageNet dataset [1], using the weights provided by PyTorchpytorch 1.6.0. The input images are resized to 224×224 dimension and rescaled to $[0, 1]$, as originally done for the ImageNet dataset. The output of the fourth layer are reshaped into 47×512 matrices. On top of the matrix S we first apply 16 convolutional filters with kernel size 5×5, then a max pooling operation with kernel size 4×4, and finally 32 convolutional filters with kernel size 5×5. We obtain a final vector of 1570 features. The final 2-layers FC module applied to the concatenation of the features is composed by a first layer producing 512 output units, followed by 0.5 dropout, then the last layer produces 2 output units. We initialize the weights of the two convolutional layers and the final two FC layers with Xavier initialization [3]. Training is done in the same way as for the SignatureMatching model, using learning rate 10^{-4} and batch size 128 in the cheque use case and 16 in the contracts use case and for public datasets.

[1]https://github.com/pytorch/vision/tree/main/torchvision/models

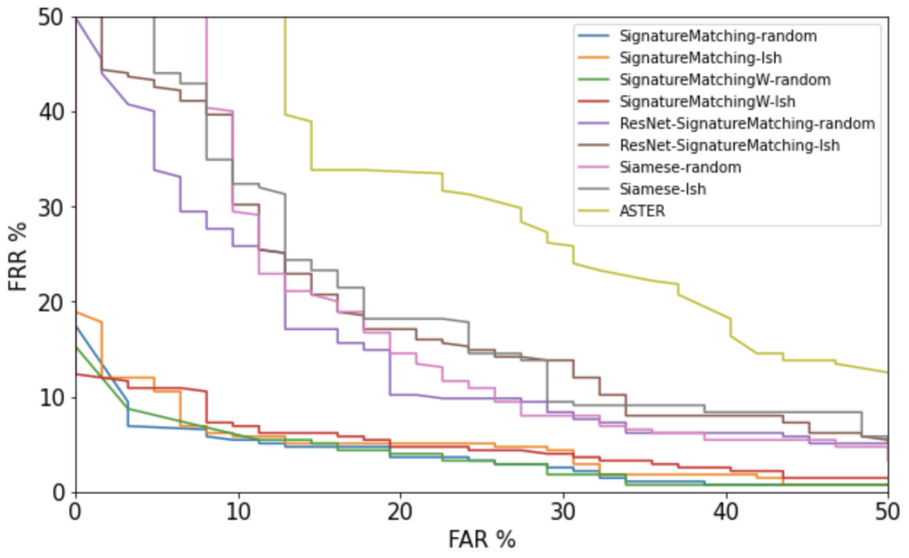


Figure 1. FAR versus FRR for a real test set of the cheque use case.

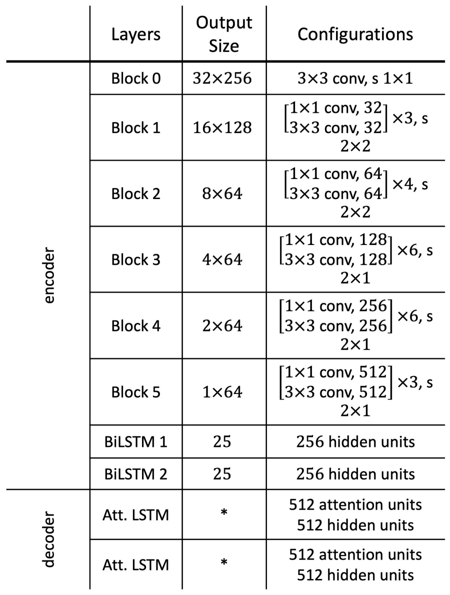
1.1.4 SignatureSiamese Model

We resize each grayscale image to 32×256 pixels and normalize pixel values to $[-1.0, 1.0]$. The image embedding part is the encoder of ASTER [5] with a final bidirectional LSTM with 256 hidden units, which produces an image embedding of dimension 64×512. The encoder is initialized with the weights of the pretrained text recognition model and then fine tuned together with all other weights in the overall architecture. The image embeddings J_1 and J_2 are flattened into vectors of dimension $64 \cdot 512 = 32768$. Then, a small FC module is applied. The first layer produces in output 1024 units, followed by dropout. Then, the final FC layer outputs 128 units. The weights of these two FC layers are initialized with Xavier initialization [3].

The model is trained with Adam optimizer, with learning rate 10^{-4} using fuzzy factor equal to 10^{-8} and weight decay equal to 0.0005. The batch size is 128 in the cheque use case. Every 20 steps the learning rate is decreased by $\gamma = 0.1$. The model is trained with the contrastive loss in Eq. (2) with $m = 1$ and $\alpha = 1$.

1.1.5 Text Recognition Model

We employ as text-recognition model the available source code of ASTER[2] for the competitor and to initialize the image encoder of the SignatureMatching models. ASTER is initialized with the weights of the publicly available pretrained model. All hyperparameters are set to the default values, except for the batch size set to 64, the height of input images set to 32, and the maximum number of epochs set to 35. The details of the architecture employed in our experiments are depicted in Fig. 2.



	Layers	Output Size	Configurations
encoder	Block 0	32×256	3×3 conv, s 1×1
	Block 1	16×128	$\begin{bmatrix}1\times1\text{ conv}, 32\\3\times3\text{ conv}, 32\end{bmatrix}\times3$, s 2×2
	Block 2	8×64	$\begin{bmatrix}1\times1\text{ conv}, 64\\3\times3\text{ conv}, 64\end{bmatrix}\times4$, s 2×2
	Block 3	4×64	$\begin{bmatrix}1\times1\text{ conv}, 128\\3\times3\text{ conv}, 128\end{bmatrix}\times6$, s 2×1
	Block 4	2×64	$\begin{bmatrix}1\times1\text{ conv}, 256\\3\times3\text{ conv}, 256\end{bmatrix}\times6$, s 2×1
	Block 5	1×64	$\begin{bmatrix}1\times1\text{ conv}, 512\\3\times3\text{ conv}, 512\end{bmatrix}\times3$, s 2×1
	BiLSTM 1	25	256 hidden units
	BiLSTM 2	25	256 hidden units
decoder	Att. LSTM	*	512 attention units 512 hidden units
	Att. LSTM	*	512 attention units 512 hidden units

Figure 2. Configuration of ASTER model used in our experiments.

1.2. Additional Experiments

In Fig. 1 we report the trade off between FRR and FAR on a small real test set available for the cheque use case for all models with real non matching pairs. Our proposed SignatureMatching models are always below the other competitors and are able to reach satisfactory performances, with FAR lower than 2% and FRR lower than 20%.

[2]https://github.com/ayumiymk/aster.pytorch

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 1

[2] Sounak Dey, Anjan Dutta, J Ignacio Toledo, Suman K Ghosh, Josep Lladós, and Umapada Pal. Signet: Convolutional siamese network for writer independent offline signature verification. *arXiv preprint arXiv:1707.02131*, 2017. 1

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010. 1, 2

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[5] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018. 1, 2