A Deep-Unfolded Reference-Based RPCA Network For Video Foreground-Background Separation

Huynh Van Luong*†, Boris Joukovsky*†, Yonina C. Eldar‡, Nikos Deligiannis*†

*Department of Electronics and Informatics, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussels, Belgium
†imec, Kapeldreef 75, B-3001 Leuven, Belgium
‡Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 7610001, Israel

***Abstract*—Deep unfolded neural networks are designed by unrolling the iterations of optimization algorithms. They can be shown to achieve faster convergence and higher accuracy than their optimization counterparts. This paper proposes a new deep-unfolding-based network design for the problem of Robust Principal Component Analysis (RPCA) with application to video foreground-background separation. Unlike existing designs, our approach focuses on modeling the temporal correlation between the sparse representations of consecutive video frames. To this end, we perform the unfolding of an iterative algorithm for solving reweighted ℓ_1-ℓ_1 minimization; this unfolding leads to a different proximal operator (a.k.a. different activation function) adaptively learned per neuron. Experimentation using the moving MNIST dataset shows that the proposed network outperforms a recently proposed state-of-the-art RPCA network in the task of video foreground-background separation.**

***Index Terms*—Deep unfolding, deep learning, robust PCA, video analysis, foreground-background separation.**

I. Introduction

Principal component analysis (PCA) [1] has been a key method for data analysis with a plethora of applications in anomaly detection, dimensionality reduction, and signal compression among many. The solution of PCA—namely, the set of orthogonal basis vectors (the principal components) that define a subspace where the data lives—can be easily obtained by applying the singular vector decomposition (SVD) on a matrix $\mathbf{M}$ formed by the data vectors. Robust PCA (RPCA) [2] is a variant of PCA that addresses the sensitivity of SVD to outliers. RPCA decomposes the data matrix $\mathbf{M}$ into the sum of a low-rank component $\mathbf{L}$, whose subspace defines the principal components, and a sparse component $\mathbf{S}$, which captures the outliers.

RPCA has found various applications, including anomaly detection for networks [3] (e.g., computer networks and social media networks), reconstruction of dynamic magnetic resonance imaging (MRI) [4], and data visualization [5]. In video analysis, which is the domain this paper focuses on, RPCA has been used to decompose a sequence of vectorized frames, comprising the columns of $\mathbf{M}$, into the background modeled by the low-rank component $\mathbf{L}$, and the foreground modeled by the sparse innovation component $\mathbf{S}$. A similar decomposition was used in ultrasound to separate tissues from blood flow [6], [7], [8].

Several optimization-based methods have been proposed to solve the low-rank plus sparse matrix decomposition problem. Candés *et al.* [2] suggested a convex formulation of the problem—referred to as principal component pursuit (PCP)—by using the ℓ_1-norm and the nuclear-norm to encode the structure of the sparse and low-rank component, respectively. Furthermore, with the goal of achieving faster convergence, non-convex methods have been proposed based on alternating minimization [9] and projected gradient descent [10]. The study in [11] introduced a memory-efficient Robust PCA and its online version achieves nearly-optimal memory complexity. We refer to [12] for a comprehensive overview.

Deep unfolding methods have been achieving state-of-the-art performance in solving decomposition problems in terms of both accuracy and computational complexity [13], [14], [15], [16], [6], [7], [8]. The study in [13] proposed to unroll the iterations of the Iterative Shrinkage-Thresholding Algorithm (ISTA) to a feed-forward neural network—coined Learned ISTA (LISTA)—which is trained on data. The learned convolutional sparse coding network in [14] and the deep-unfolded recurrent neural network in [15] are convolutional and recurrent extensions of LISTA that solve the convolutional and the dynamic sparse representation problem, respectively. Regarding the low-rank-plus-sparse decomposition problem, the authors of [16] unrolled the iterations of proximal gradient methods to form a deep feed-forward neural network. Furthermore, the works [6], [7], [8] proposed the Deep Convolutional Robust PCA network (CORONA), which uses convolutional layers and an SVD for the low-rank approximation.

In this paper, we present a deep unfolded RPCA network for the problem of video foreground-background separation. Our design aims to capture the inherent temporal correlation among the sparse representations of consecutive video frames. To this end, we propose a RPCA network that unfolds an iterative algorithm for solving reweighted ℓ_1-ℓ_1 minimization [17], an extension of reweighted ℓ_1 minimization [18]. Our unfolded design leads to a new proximal operator with multiple thresholds (a.k.a. activation functions), which are adaptively learnt per neuron, thereby increasing network adap-

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (Projects G040016N and G0A4720N) and from Innoviris (Project ADVISE), Belgium. This work was also supported by the European Union's Horizon 2020 research and innovation program (Grant 646804-ERC-COG-BNYQ).

tivity and expressivity. Experimentation on the moving MNIST dataset [19] shows that the proposed network outperforms the CORONA [6] network in terms of accuracy and convergence speed.

The rest of the paper is as follows: Section II presents the background and Section III describes the proposed refRPCA network. The experimental evaluation of our model is given in Section IV, whereas Section V draws the conclusion.

II. Background on RPCA

In this section, we review existing optimization-based methods [20], [21], [22] and deep-learning-based models [6] addressing the RPCA problem.

A. Optimization-Based Methods for RPCA

Traditional methods for RPCA [20], [21], [22] decompose the data matrix $\mathbf{M}$ into $\mathbf{S}$ and $\mathbf{L}$ by solving the principal component pursuit (PCP) [21] problem:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \text{ s.t. } \mathbf{M} = \mathbf{L} + \mathbf{S}, \tag{1}$$

where $\|\mathbf{L}\|_* = \sum_i \sigma_i(\mathbf{L})$ is the nuclear norm—sum of singular values $\sigma_i(\mathbf{L})$—of the matrix $\mathbf{L}$, $\|\mathbf{S}\|_1 = \sum_{i,j} |\mathrm{s}_{i,j}|$ is the ℓ_1-norm of $\mathbf{S}$ organized in a vector, and λ is a regularization parameter. The aforementioned RPCA methods [20], [21], [22] typically assume that the $\mathbf{L}$ component lies in a low-dimensional subspace, i.e., the background frames are static or slowly-changing. In video foreground-background separation, a sequence of m vectorized frames (modeled by $\mathbf{M} \in \mathbb{R}^{n\times m}$) is separated into the slowly-changing background $\mathbf{L}$ and the sparse foreground $\mathbf{S}$.

Problem (1) can be formulated in a Lagrangian form

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F^2 + \lambda_1\|\mathbf{L}\|_* + \lambda_2\|\mathbf{S}\|_1, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenious norm and λ_1 and λ_2 are tuning parameters. By using proximal gradient methods [23] to solve (2), $\mathbf{L}^{(k+1)}$ and $\mathbf{S}^{(k+1)}$ at iteration k+1 can be iteratively computed via the singular value thresholding operator [24] for $\mathbf{L}$ and the soft thresholding operator [23] for $\mathbf{S}$.

B. Deep Unfolding for RPCA

The deep convolutional RPCA (CORONA) network in [7] considered measurement matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ for the $\mathbf{L}$ and $\mathbf{S}$ components, respectively. The decomposition problem was then formulated as

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}\|\mathbf{M} - \mathbf{H}_1\mathbf{L} - \mathbf{H}_2\mathbf{S}\|_F^2 + \lambda_1\|\mathbf{L}\|_* + \lambda_2\|\mathbf{S}\|_{1,2}, \tag{3}$$

where $\|\cdot\|_{1,2}$ is the mixed $\ell_{1,2}$ norm. Problem (3) was solved via iteratively updating $\mathbf{L}^{(k+1)}$ and $\mathbf{S}^{(k+1)}$ at iteration $k+1$ with

$$\mathbf{L}^{(k+1)} = \Gamma_{\frac{\lambda_1}{c}}\left(\left(\mathbf{I} - \frac{1}{c}\mathbf{H}_1^T\mathbf{H}_1\right)\mathbf{L}^{(k)} - \mathbf{H}_1^T\mathbf{H}_2\mathbf{S}^{(k)} + \mathbf{H}_1^T\mathbf{M}\right) \tag{4a}$$

$$\mathbf{S}^{(k+1)} = \Phi_{\frac{\lambda_2}{c}}\left(\left(\mathbf{I} - \frac{1}{c}\mathbf{H}_2^T\mathbf{H}_2\right)\mathbf{S}^{(k)} - \mathbf{H}_2^T\mathbf{H}_1\mathbf{L}^{(k)} + \mathbf{H}_2^T\mathbf{M}\right), \tag{4b}$$

where $\Gamma_{\frac{\lambda_1}{c}}(.)$ and $\Phi_{\frac{\lambda_2}{c}}(.)$ are the singular value thresholding [24] and mixed $\ell_{1,2}$ soft thresholding [23] operators, respectively, and c is a Lipschitz constant.

Following the principles of deep unfolding, the authors of [6] proposed to unroll the iterations of the algorithm solving Problem (3) into a multiple-layer neural network, the k^{th} layer of which computes:

$$\mathbf{L}^{(k+1)} = \Gamma_{\lambda_1^{(k)}}\left\{\mathbf{W}_1^{(k)} * \mathbf{M} + \mathbf{W}_3^{(k)} * \mathbf{S}^{(k)} + \mathbf{W}_5^{(k)} * \mathbf{L}^{(k)}\right\} \tag{5a}$$

$$\mathbf{S}^{(k+1)} = \Phi_{\lambda_2^{(k)}}\left\{\mathbf{W}_2^{(k)} * \mathbf{M} + \mathbf{W}_4^{(k)} * \mathbf{S}^{(k)} + \mathbf{W}_6^{(k)} * \mathbf{L}^{(k)}\right\}, \tag{5b}$$

where $*$ denotes the convolution operator. In CORONA, the weights of the convolutional layers $\mathbf{W}_1^{(k)}, \cdots, \mathbf{W}_6^{(k)}$ and the regularization parameters $\lambda_1^{(k)}$, $\lambda_2^{(k)}$ are learned from training data using back-propagation.

III. The Proposed refRPCA-Net Network

A. The Proposed Unfolded Method

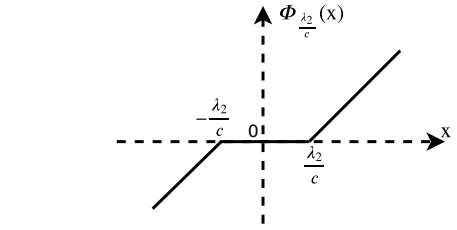
We consider the problem of video foreground-background separation and attempt to capture the inherent temporal correlation in video. We assume that the foregrounds of two consecutive frames $\mathbf{s}_{t-1}, \mathbf{s}_t$ in $\mathbf{S} = [\mathbf{s}_1, \cdots, \mathbf{s}_m]$ are correlated via a projection matrix $\mathbf{P}$, under the temporal correlation assumption, i.e., $\mathbf{s}_t \approx \mathbf{P}\mathbf{s}_{t-1}$. For simplicity, $\mathbf{P}$ is not varying across time. Then, we construct a sequence of reference frames $\mathbf{S_P}$ by $\mathbf{S_P} = [\mathbf{s}_1, \mathbf{P}\mathbf{s}_1, \cdots, \mathbf{P}\mathbf{s}_{m-1}]$. We propose a reference-based RPCA (refRPCA) problem that leverages the correlated reference $\mathbf{S_P}$ of the sparse component $\mathbf{S}$ via ℓ_1-ℓ_1 minimization [25], [17] to improve the separation problem in (3). Furthermore, the proposed refRPCA uses a reweighting scheme [18], [17] via reweighting the elements of $\mathbf{S}$ with a matrix $\mathbf{Q}$. The latter choice results from the observation that reweighted ℓ_1-ℓ_1-minimization [17] outperforms its non-reweighted counterpart, leading to more accurate sparse representations. Thus, the refRPCA problem is formulated as

$$\begin{aligned}\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}\|\mathbf{M} - \mathbf{H}_1\mathbf{L} - \mathbf{H}_2\mathbf{S}\|_F^2 + \lambda_1\|\mathbf{L}\|_* \\ + \lambda_2\|\mathbf{Q}\circ\mathbf{S}\|_1 + \lambda_3\|\mathbf{Q}\circ(\mathbf{S} - \mathbf{S_P})\|_1,\end{aligned} \tag{6}$$

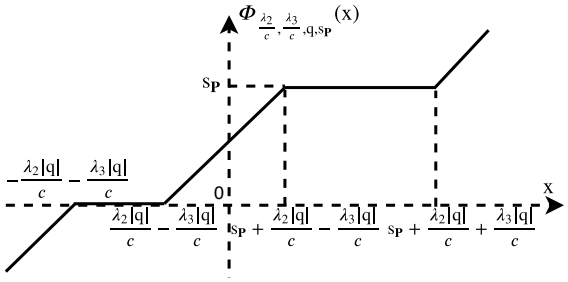
where "$\circ$" denotes element-wise multiplication and a weighting matrix $\mathbf{Q} \in \mathbb{R}^{n\times m}$ is defined as $\mathbf{Q} = [\mathbf{q}, \cdots, \mathbf{q}]$ with $\mathbf{q} \in \mathbb{R}^n$, which consists of m weighting vectors $\mathbf{q}$.

We solve (6) using a proximal gradient method [23], where the low-rank component $\mathbf{L}^{(k+1)}$ is iteratively computed via the singular value thresholding operator [24] as in (4a). The sparse component $\mathbf{S}^{(k+1)}$ is updated using a new proximal operator $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathbf{q},\mathbf{S_P}}(.)$, which is formulated for the reweighted-ℓ_1-ℓ_1 minimization [18], [17], that is,

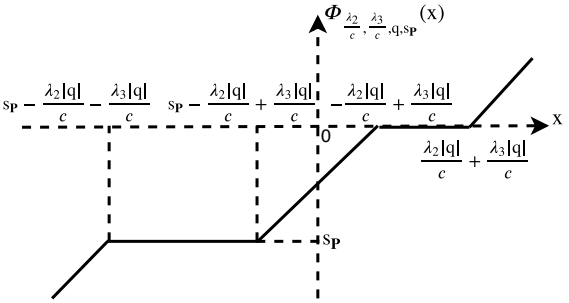
$$\begin{aligned}\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathbf{q},\mathbf{S_P}}(\mathbf{X}) = \arg\min_{\mathbf{U}} \Big\{ \frac{\lambda_2}{c}\|\mathbf{Q}\circ\mathbf{U}\|_1 \\ + \frac{\lambda_3}{c}\|\mathbf{Q}\circ(\mathbf{U} - \mathbf{S_P})\|_1 + \frac{1}{2}\|\mathbf{U} - \mathbf{X}\|_2^2\Big\}.\end{aligned} \tag{7}$$



(a) CORONA: $\Phi_{\frac{\lambda_2}{c}}(\mathrm{x})$



(b) refRPCA: $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c}\mathrm{q},\mathrm{s_P}}(\mathrm{x})$ for $\mathrm{s_P}\geq 0$.



(c) refRPCA: $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathrm{q},\mathrm{s_P}}(\mathrm{x})$ for $\mathrm{s_P}<0$.

Fig. 1. The proximal operators of CORONA [7] (a) vs refRPCA (b) and (c). The parameters λ_2, λ_3 and c are learned globally. Note that (b) and (c) are drawn for given q and $\mathrm{s_P}$. The weight q allows for a different proximal operator for each entry of x due to the varying length of the multiple-threshold intervals. The reference $\mathrm{s_P}$ defines the position of the non-zero plateau each time the operator is evaluated.

Since Problem (7) is separable, it can be formulated element-wise. Let q, $\mathrm{s_P}$, x, u denote each element of the corresponding $\mathbf{Q}$, $\mathbf{S_P}$, $\mathbf{X}$, $\mathbf{U}$. Then,

$$\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathrm{q},\mathrm{s_P}}(\mathrm{x}) = \arg\min_{\mathrm{u}} \Big\{ \frac{\lambda_2}{c}|\mathrm{qu}| + \frac{\lambda_3}{c}|\mathrm{q}(\mathrm{u}-\mathrm{s}_p)| + \frac{1}{2}(\mathrm{u}-\mathrm{x})^2 \Big\}. \tag{8}$$

The solution of (8) is derived in [17] and is given as follows. For $\mathrm{s_P}\geq 0$:

$$\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathrm{q},\mathrm{s_P}}(\mathrm{x}) = \begin{cases} \mathrm{x}-\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c}, & \mathrm{s_P}+\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} < \mathrm{x} < \infty \\ \mathrm{s_P}, & \mathrm{s_P}+\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c} \leq \mathrm{x} \leq \mathrm{s_P}+\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} \\ \mathrm{x}-\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c}, & \frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c} < \mathrm{x} < \mathrm{s_P}+\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c} \\ 0, & -\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c} \leq \mathrm{x} \leq \frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c} \\ \mathrm{x}+\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c}, & -\infty < \mathrm{x} < -\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c}, \end{cases} \tag{9}$$

Algorithm 1: The proposed refRPCA algorithm for foreground-background separation.

1 **Input:** $\mathbf{M}$, $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{H}_1$, $\mathbf{H}_2$, λ_1, λ_2, λ_3, c, the maximum number of iterations d.
2 **Output:** $\widehat{\mathbf{L}}$, $\widehat{\mathbf{S}}$.
3 **for** *k = 1 to d* **do**
4 $\quad \widetilde{\mathbf{L}}^{(k)} = \Big(\mathbf{I} - \frac{1}{c}\mathbf{H}_1^T\mathbf{H}_1\Big)\mathbf{L}^{(k)} - \mathbf{H}_1^T\mathbf{H}_2\mathbf{S}^{(k)} + \mathbf{H}_1^T\mathbf{M}$
5 $\quad \widetilde{\mathbf{S}}^{(k)} = \Big(\mathbf{I} - \frac{1}{c}\mathbf{H}_2^T\mathbf{H}_2\Big)\mathbf{S}^{(k)} - \mathbf{H}_2^T\mathbf{H}_1\mathbf{L}^{(k)} + \mathbf{H}_2^T\mathbf{M}$
6 $\quad \mathbf{L}^{(k+1)} = \Gamma_{\frac{\lambda_1}{c}}\Big(\widetilde{\mathbf{L}}^{(k)}\Big)$
7 $\quad \mathbf{S_P} = [\mathbf{s}_1^{(k)}, \mathbf{P}\mathbf{s}_1^{(k)}, \cdots, \mathbf{P}\mathbf{s}_{m-1}^{(k)}]$
8 $\quad \mathbf{S}^{(k+1)} = \Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathbf{q},\mathbf{S_P}}\Big(\widetilde{\mathbf{S}}^{(k)}\Big)$
9 **end**
10 **return** $\widehat{\mathbf{L}} = \mathbf{L}^{(k+1)}$, $\widehat{\mathbf{S}} = \mathbf{S}^{(k+1)}$.

and for $\mathrm{s_P}<0$:

$$\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c}\mathrm{q},\mathrm{s_P}}(\mathrm{x}) = \begin{cases} \mathrm{x}-\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c}, & \frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} < \mathrm{x} < \infty \\ 0, & -\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} \leq \mathrm{x} \leq \frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} \\ \mathrm{x}+\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c}, & \mathrm{s_P}-\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} < \mathrm{x} < -\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} \\ \mathrm{s_P}, & \mathrm{s_P}-\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c} \leq \mathrm{x} \leq \mathrm{s_P}-\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c} \\ \mathrm{x}+\frac{\lambda_2|\mathrm{q}|}{c}+\frac{\lambda_3|\mathrm{q}|}{c}, & -\infty < \mathrm{x} < \mathrm{s_P}-\frac{\lambda_2|\mathrm{q}|}{c}-\frac{\lambda_3|\mathrm{q}|}{c}. \end{cases} \tag{10}$$

The proposed refRPCA leads to the proximal operator $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathrm{q},\mathrm{s_P}}(.)$ in (8), which replaces $\Phi_{\frac{\lambda_2}{c}}(.)$ (4b) in CORONA [7]. The difference is illustrated in Fig. 1: Fig. 1(a) shows the proximal operator for CORONA; Figs. 1(b) and 1(c) depict the generic form of the proximal operator for $\mathrm{s_P}\geq 0$ [Fig. 1(b)] and $\mathrm{s_P}<0$ [Fig. 1(c)]. Observe that the proximal function for CORONA [Fig. 1(a)] has a single threshold, while, for refRPCA [Fig. 1(b) and Fig. 1(c)], different values of q lead to different shapes of the proximal functions $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathrm{q},\mathrm{s_P}}(.)$ for each entry of the input, due to the varying length of the multiple-threshold intervals.

The algorithm for solving Problem (6) is summarized in Algorithm 1, where $\Phi_{\frac{\lambda_2}{c},\frac{\lambda_3}{c},\mathbf{q},\mathbf{S_P}}(.)$ in Line 8 is given by (9) and (10). As shown in Algorithm 1, we need to properly tune several parameters: $\mathbf{P}$, $\mathbf{Q}$, λ_1, λ_2, λ_3, which play significant roles in the singular value thresholding and proximal operators [see Lines 6 and 8].

B. The refRPCA-Net Architecture

We now propose to unroll the iterations of the refRPCA algorithm into an d-layer network, coined *deep-unfolded reference-based RPCA network* (refRPCA-Net). The k^{th} iteration in Algorithm 1 corresponds to the k^{th} layer in refRPCA-Net, which is illustrated in Fig. 2. At each layer k, the low-

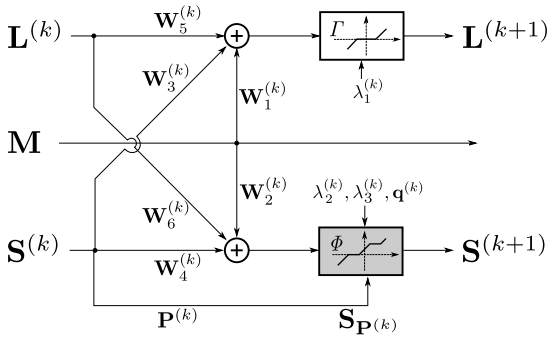


Fig. 2. The proposed refRPCA-Net architecture.

rank component $\mathbf{L}^{(k+1)}$ is updated as in (5a), while $\mathbf{S}^{(k+1)}$ is updated as

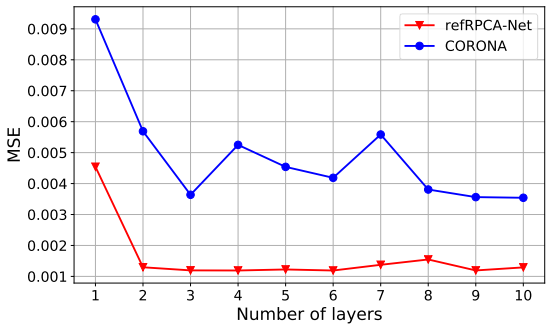
$$\mathbf{S}^{(k+1)} = \Phi_{\lambda_2^{(k)},\lambda_3^{(k)},\mathbf{q}^{(k)},\mathbf{S}_{\mathbf{P}^{(k)}}} \Big\{ \mathbf{W}_2^{(k)} * \mathbf{M} + \mathbf{W}_4^{(k)} * \mathbf{S}^{(k)} + \mathbf{W}_6^{(k)} * \mathbf{L}^{(k)} \Big\}, \tag{11}$$

where the parameters $\lambda_1^{(k)}, \lambda_2^{(k)}, \lambda_3^{(k)}, \mathbf{q}^{(k)}, \mathbf{P}^{(k)}$ and the parameters of the convolutional layers $\mathbf{W}_1^{(k)}, \cdots, \mathbf{W}_6^{(k)}$ are learned per layer during training.

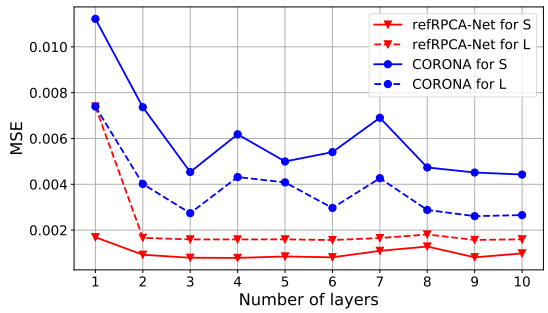
The key difference with CORONA [7] is the proximal operator (a.k.a., activation function) Φ [see the block highlighted in gray color in Fig. 2]. Φ incorporates the reference $\mathbf{S}_{\mathbf{P}^{(k)}}$ via the matrix $\mathbf{P}^{(k)}$, defining by $\mathbf{S}_{\mathbf{P}^{(k)}} = [\mathbf{s}_1^{(k)}, \mathbf{P}^{(k)}\mathbf{s}_1^{(k)}, \cdots, \mathbf{P}^{(k)}\mathbf{s}_{m-1}^{(k)}]$. Furthermore, the different learnable values of $\mathrm{q}_i \in \mathbf{q}^{(k)}$ lead to different realizations of the activation functions $\Phi_{\lambda_2^{(k)},\lambda_3^{(k)},\mathrm{q}_i,\mathbf{S}_{\mathbf{P}^{(k)}}}(.)$ for each neuron [see Figs. 1(b) and 1(c)]. This increases the adaptivity and expressivity of refRPCA-Net. We note that the low-rank component $\mathbf{L}^{(k)}$ is updated in a similar manner as in CORONA. However, after each layer, the updated $\mathbf{S}^{(k)}$ becomes one of the inputs for updating $\mathbf{L}^{(k+1)}$ [see (5a)]. Therefore, an improvement in the estimation of $\mathbf{S}$ leads to an improvement in the estimation of $\mathbf{L}$ and vice versa.

IV. Experiments

In this section we assess the performance of the proposed refRPCA-Net versus CORONA [7] in the task of video foreground-background separation. We consider the moving MNIST dataset [19] for our experiments, which contains 10,000 sequences of moving digits, each 20 frames long. We resize each frame in the dataset to a resolution of 32×32 pixels, thereby having $m = 20$ and $n = 1024$ according to our notation (see Section III). We add a synthetic low-rank background to each sequence that is generated as in [17]; namely, we generate $\mathbf{L} \doteq \mathbf{U}\mathbf{V}^T$, with $\mathbf{U} \in \mathbb{R}^{n\times r}$ and $\mathbf{V} \in \mathbb{R}^{m\times r}$ sampled from a standard Gaussian distribution and the rank set to $r = 5$. The dataset is then split into 8000, 1000, and 1000 samples respectively for training, validation, and testing. The pixel intensities are normalized to the unit range before being fed to the network. The two models are trained using the Adam optimizer with a learning rate of 10^{-3},



(a) Average MSE.



(b) MSE for $\mathbf{L}$ and $\mathbf{S}$.

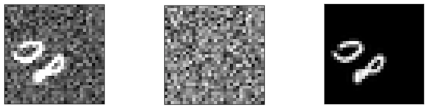
Fig. 3. Average Mean Square Error (MSE) vs. the number of layers for the proposed refRPCA-Net and CORONA [6] on video separation.

a batch size of 200 and during 50 epochs by minimizing the following compound mean square error (MSE) loss:

$$\mathcal{L}(\Theta) = \frac{1}{2N}\sum_{i=1}^{N} \|\mathbf{L}_i - \widehat{\mathbf{L}}_i\|_F^2 + \frac{1}{2N}\sum_{i=1}^{N} \|\mathbf{S}_i - \widehat{\mathbf{S}}_i\|_F^2, \tag{12}$$

where $\Theta = \Big\{\mathbf{W}_1^{(k)}, \cdots, \mathbf{W}_6^{(k)}, \lambda_1^{(k)}, \lambda_2^{(k)}, \lambda_3^{(k)}, \mathbf{q}^{(k)}, \mathbf{P}^{(k)}\Big\}_{k=1}^{d}$ are the learnt parameters, $\{\mathbf{S}_i, \mathbf{L}_i\}_{i=1}^N$ are the ground-truth and $\{\widehat{\mathbf{S}}_i, \widehat{\mathbf{L}}_i\}_{i=1}^N$ the foreground and background components predicted by the network (with $\mathbf{S}_i, \mathbf{L}_i, \widehat{\mathbf{S}}_i, \widehat{\mathbf{L}}_i \in \mathbb{R}^{1024\times 20}$), and N is the number of training samples in the dataset. Both networks are trained with different number of layers, that is, from 1 to 10 layers. The remaining network configurations, including the convolutional kernel sizes and the initialization of the weights, are kept identical to what is reported in [6] to ensure a fair comparison.

We plot the reconstruction mean squared-error (MSE)—averaged over the sequences in the validation set—versus the number of layers for each model; specifically, Fig. 3(a) reports the average MSE of both components whereas Fig. 3(b) reports the average MSE for the low-rank and sparse components separately. We observe that the proposed refRPCA-Net outperforms CORONA. Similar to what has been reported in [7], for both networks, increasing the number of layers above a certain number does not lead to a significant performance improvement, while inducing higher complexity in training and inference. The proposed network, however, achieves faster convergence and delivers more stable performance to the

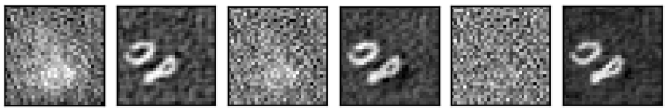


(a) (b) (c)

Fig. 4. (a) Original frame, Ground-truth (b) background and (c) foreground.

(a) 1 layer. (b) 1 layer. (c) 2 layers. (d) 2 layers. (e) 10 layers. (f) 10 layers.

Fig. 5. Visual results for refRPCA-Net for different number of layers: (a), (c), (e) Background frames and (b), (d), (f) Foreground frames.



(a) 1 layer. (b) 1 layer. (c) 2 layers. (d) 2 layers. (e) 10 layers. (f) 10 layers.

Fig. 6. Visual results for CORONA for different number of layers: (a), (c), (e) Background frames and (b), (d), (f) Foreground frames.

number of layers than CORONA. Additionally, a foreground-background separation example is displayed in Fig. 5 for refRPCA-Net and Fig. 6 for CORONA, for the 1, 2 and 10 layers configurations. Again, we observe that refRPCA-Net leads to better estimates of the sparse and low-rank components compared to CORONA.

The performance gain of refRPCA-Net compared to CORONA is higher for the sparse component because the method directly leverages the reference $\mathbf{S}_{\mathbf{P}^{(k)}}$ through the reweighted activation function. Furthermore, while the two networks use the same update rule for the low-rank component, the gain in the sparse component estimation in refRPCA-Net indirectly leads to an improved estimation of the low-rank component as well. This gain is notable when the number of network layers is higher than 1. When the number of layers is set to 1 (corresponding to one iteration of the method), the enhanced estimation of the sparse component is not exploited to improve the reconstruction of the low-rank component.

V. Conclusion

We proposed a deep-unfolded reference-based RPCA network for the task of video foreground-background separation. Our refRPCA-Net architecture captures the correlation between the sparse components of consecutive video frames by unfolding an algorithm for solving reweighted ℓ_1-ℓ_1 minimization. Our design leads to a different proximal operator (activation function) adaptively learnt per neuron. Our experiments using the moving MNIST dataset show that our network outperforms the state-of-the-art CORONA network consistently for the reconstruction of both the foreground and background components.

References

[1] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent lab. sys.*, vol. 2, no. 1-3, pp. 37–52, 1987.

[2] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

[3] M. Mardani, G. Mateos, and G. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 50–66, 2012.

[4] R. Otazo, E. Candés, and D. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[5] Konstantinos Slavakis, Georgios B Giannakis, and Gonzalo Mateos, "Modeling and optimization for big data analytics: (Statistical) learning tools for our era of data deluge," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, 2014.

[6] R. Cohen, Y. Zhang, O. Solomon, D. Toberman, L. Taieb, R. J. van Sloun, and Y. C. Eldar, "Deep convolutional robust pca with application to ultrasound imaging," in *ICASSP 2019*, May 2019.

[7] O. Solomon, R. Cohen, Y. Zhang, Y. Yang, Q. He, J. Luo, R. van Sloun, and Y. Eldar, "Deep unfolded robust pca with application to clutter suppression in ultrasound," *IEEE Trans. on medical imaging*, 2019.

[8] R. J. G. van Sloun, R. Cohen, and Y. C. Eldar, "Deep learning in ultrasound imaging," *Proceedings of the IEEE*, vol. 108, no. 1, pp. 11–29, Jan 2020.

[9] P. Netrapalli, UN Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *NIPS*, 2014, pp. 1107–1115.

[10] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *NIPS*, 2016, pp. 4152–4160.

[11] P. Narayanamurthy and N. Vaswani, "A fast and memory-efficient algorithm for robust pca (merop)," in *2018 IEEE ICASSP*, 2018.

[12] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery," *IEEE signal processing magazine*, vol. 35, no. 4, pp. 32–55, 2018.

[13] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding," in *Proc. of the 27th International Conference on Machine Learning*, 2010, ICML'10, pp. 399–406.

[14] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *2018 IEEE ICASSP*, April 2018.

[15] H. D. Le, H. Van Luong, and N. Deligiannis, "Designing recurrent neural networks by unfolding an l1-l1 minimization algorithm," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019.

[16] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1821–1833, Sep. 2015.

[17] H. Van Luong, N. Deligiannis, J. Seiler, S. Forchhammer, and A. Kaup, "Compressive online robust principal component analysis via n-ℓ_1 minimization," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4314–4329, 2018.

[18] Emmanuel J. Candès, Michael B. Wakin, and Stephen P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[19] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[20] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *NIPS*, 2009.

[21] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11:1–11:37, June 2011.

[22] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[23] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2(1), pp. 183–202, 2009.

[24] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[25] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Compressed sensing with prior information: Strategies, geometry, and bounds," *IEEE Trans. on Information Theory*, vol. 63, no. 7, pp. 4472–4496, July 2017.