The focus of this paper is speeding up the application of convolutional neural networks (CNNs). While delivering impressive results across a range of computer vision and machine learning tasks, these networks are computationally demanding, limiting their deployability. Convolutional layers generally consume the bulk of the processing time, and so in this work we present two simple schemes for drastically speeding up these layers. This is achieved by exploiting cross-channel or filter redundancy to construct a low rank basis of filters that are rank-1 in the spatial domain. Our methods are architecture agnostic, and can be easily applied to existing CPU and GPU convolutional frameworks for tuneable speedup performance. We demonstrate this with a real world network designed for scene text character recognition [1], showing a possible 2.5 $\times$ speedup with no loss in accuracy, and 4.5 $\times$ speedup with less than 1% drop in accuracy, still achieving state-of-the-art on standard benchmarks.

**Approximation Schemes.** We provide the frameworks for two methods to approximate the $N$ 3D filters $W_n$ of a convolutional layer acting on the input $z$ with $C$ channels $z^c_n$ such that $W_n * z = \sum_{c=1}^{C} W_n^c * z^c_n$. Both methods exploit the redundancy that exists between different feature channels and filters of convolutional layers.

**Scheme 1** builds upon the work of Rigamonti et al. [2] and approximates the original set of full-rank filters as a linear combination of a smaller set of separable (rank-1) filters. The separability of these filters allows convolutions to be computed much more efficiently than the full-rank filters by splitting the full convolution into horizontal convolutions followed by vertical convolution. For a layer of $N$ convolutional filters $W_n$, $n \in [1 \ldots N]$, where each filter acts on a single channel $c$ of the 3D input, $c \in [1 \ldots C]$, we learn a basis of $M$ separable filters $s^c_m$, $m \in [1 \ldots M]$, where $M < N$, as well as the coefficients $a^c_m$ to linearly combine them, such that the original filter $W_n^c \approx \sum_{m=1}^{M} a^c_m s^c_m$ offering a speedup due to the separable convolution and smaller basis of filters required for convolution.

**Scheme 2** also employs the idea of separable convolutions, but uses a separate basis of vertical filters and horizontal filters. The original convolutional layer is approximated by a vertical convolution layer with $K$ vertical filters $v_k : k \in [1 \ldots K]$ followed by a horizontal convolutional layer with horizontal filters $h_n : n \in [1 \ldots N]$. This results in the original filters being approximated by the sequence of these two layers, i.e. $W_n^c \approx \sum_{k=1}^{K} h_n^k * v_k$. The advantage of this method over Scheme 1 is that it can be plugged straight in to any CNN toolbox that supports rectangular filters.

For both schemes, the speedup can be tuned by varying the number of filters for each basis.

**Optimization.** The separable approximations can be optimized using two methods — filter reconstruction optimization and data reconstruction optimization. Filter reconstruction optimization aims to minimise the reconstruction error of the original filters by the approximation, whereas data reconstruction optimization aims to reconstruct the output of the original layer by the approximated layer for the training data inputs, minimising the reconstruction error using traditional back-propagation of errors.

**Results.** We provide the results of our approximation schemes on scene text character recognition using the state-of-the-art classifier of [1], under different scenarios and settings. A 4.5 $\times$ speedup can be obtained with virtually no loss in classifier accuracy (Fig. 2), with data reconstruction optimization improving the accuracy of both schemes.
