

# Vision SoC: AI-Based Image Super Resolution And Enhancement System

1. Mr. Ramesh Babu M. Tech  
Assistant Professor  
Department of CSE, TEC,  
[547sreepathi@gmail.com](mailto:547sreepathi@gmail.com)

2. N. Surya Vamsi  
[svnallagonda@gmail.com](mailto:svnallagonda@gmail.com)  
4. J. Abishek  
[abhishekjanga82@gmail.com](mailto:abhishekjanga82@gmail.com)

3. N. Divija  
[divijanallamothu2294@gmail.com](mailto:divijanallamothu2294@gmail.com)  
5. Surya Kiran  
[suryakiranmundru@gmail.com](mailto:suryakiranmundru@gmail.com)

## Abstract

*Recently, several models based on deep neural networks have achieved great success in terms of both reconstruction accuracy and computational performance for single image super-resolution. In these methods, the low resolution (LR) input image is upscaled to the high resolution (HR) space using a single filter, commonly bicubic interpolation, before reconstruction. This means that the super-resolution (SR) operation is performed in HR space. We demonstrate that this is sub-optimal and adds computational complexity. In this paper, we present the first convolutional neural network (CNN) capable of real-time SR of 1080p videos on a single K2 GPU. To achieve this, we propose a novel CNN architecture where the feature maps are extracted in the LR space. In addition, we introduce an efficient sub-pixel convolution layer which learns an array of upscaling filters to upscale the final LR feature maps into the HR output. By doing so, we effectively replace the handcrafted bicubic filter in the SR pipeline with more complex upscaling filters specifically trained for each feature map.*

version of HR data. It is a highly ill-posed problem, due to the loss of high-frequency information that occurs during the non-invertible low-pass filtering and subsampling operations. Effectively a one-to-many mapping from LR to HR space which can have multiple solutions, of which determining the correct solution is non-trivial. A key assumption that underlies many SR techniques is that much of the high-frequency data is redundant and thus can be accurately reconstructed from low frequency components. SR is therefore an inference problem, and thus relies on our model of the statistics of images in question. Many methods assume multiple images are available as LR instances of the same scene with different perspectives, i.e. with unique prior affine transformations. These can be categorised as multi-image SR methods [1, 11] and exploit these techniques seek to learn *implicit redundancy* that is present in natural data to recover missing HR information from a single LR instance. This usually arises in the form of local spatial correlations for images and additional temporal correlations in videos. In this case, prior information in the form of reconstruction constraints is needed to restrict the solution space of the reconstruction.

## 1. Introduction

The recovery of a high resolution (HR) image or video from its low resolution (LR) counter part is topic of great interest in digital image processing. This task, referred to as super-resolution (SR), finds direct applications in many areas such as HDTV [15], medical imaging [28, 33], satellite imaging [38], face recognition [17] and surveillance [53]. The global SR problem assumes LR data to be a low-pass filtered (blurred), down sampled and noisy

and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

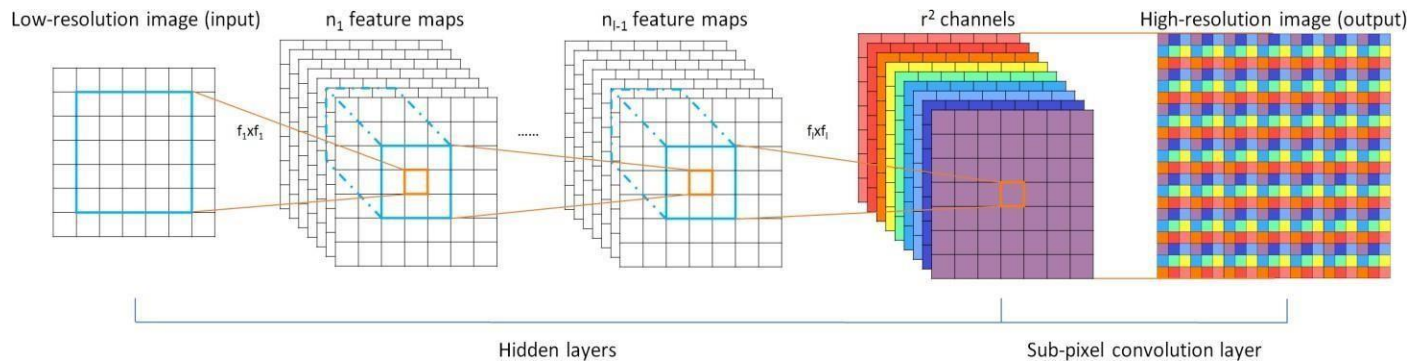


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

### 1.1. Related Work

The goal of SISR methods is to recover a HR image from a single LR input image [14]. Recent popular SISR methods can be classified into edge-based [35], image statistics-based [9, 18, 46, 12] and patch-based [2, 43, 52, 13, 54, 40, 5] methods. A detailed review of more generic SISR methods can be found in [45]. One family of approaches that has recently thrived in tackling the SISR problem is sparsity-based techniques. Sparse coding is an effective mechanism that assumes any natural image can be sparsely represented in a transform domain. This transform domain is usually a dictionary of image atoms [25, 10], which can be learnt through a training process that tries to discover the correspondence between LR and HR patches. This dictionary is able to embed the prior knowledge necessary to constrain the ill-posed problem of super-resolving unseen data. This approach is proposed in the methods of [47, 8]. A drawback of sparsity-based techniques is that introducing the sparsity constraint through a nonlinear reconstruction is generally computationally expensive. Image representations derived via neural networks [21, 49, 34] have recently also shown promise for SISR. These methods, employ the back-propagation algorithm

[22] to train on large image databases such as ImageNet [30] in order to learn nonlinear mappings of LR and HR image patches. Stacked collaborative local auto-encoders are used in [4] to super-resolve the LR image layer by layer. Osendorfer et al. [27] suggested a method for SISR based on an extension of the predictive convolutional sparse coding framework [29]. A multiple layer convolutional neural network (CNN) inspired by sparse coding methods is proposed in [7]. Chen et. al. [3] proposed

to use multi-stage trainable nonlinear reaction diffusion (TNRD) as an alternative to CNN where the weights and the nonlinearity is trainable. Wang et. al [44] trained a cascaded sparse coding network from end to end inspired by LISTA (Learning iterative shrinkage and thresholding algorithm) [16] to fully exploit the natural sparsity of images. The network structure is not limited to neural networks, for example, a random forest [31] has also been successfully used for SISR.

### 1.2. Motivations and contributions

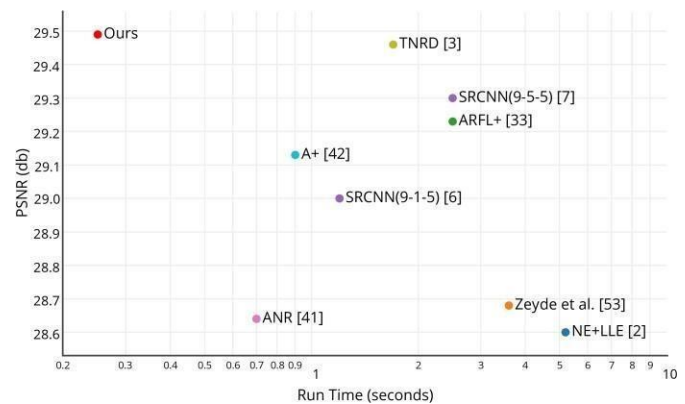


Figure 2. Plot of the trade-off between accuracy and speed for different methods when performing SR upscaling with a scale factor of 3. The results present the mean PSNR and run-time over the images from Set14 run on a single CPU core clocked at 2.0 GHz.

With the development of CNN, the efficiency of the algorithms, especially their computational and memory cost, gains importance [36]. The flexibility of deep network models to learn nonlinear relationships has been shown to attain superior reconstruction accuracy compared to previously hand-crafted models [27, 7, 44, 31, 3]. To super-resolve a LR image into HR space, it is necessary to increase the resolution of the LR image to match that

of the HR image at some point. In Osendorfer et al. [27], the image resolution is increased in the middle of the network gradually. Another popular approach is to increase the resolution before or at the first layer of the network [7, 44, 3]. However, this approach has a number of drawbacks. Firstly, increasing the resolution of the LR images before the image 1875 enhancement step increases the computational complexity. This is especially problematic for convolutional networks, where the processing speed directly depends on the input image resolution. Secondly, interpolation methods typically used to accomplish the task, such as bicubic interpolation [7, 44, 3], do not bring additional information to solve the ill-posed reconstruction problem. Learning upscaling filters was briefly suggested in the footnote of Dong et.al. [6]. However, the importance of integrating it into the CNN as part of the SR operation was not fully recognised and the option not explored. Additionally, as noted by Dong et al. [6], there are no efficient implementations of a convolution layer whose output size is larger than the input size and well- optimized implementations such as convnet [21] do not trivially allow such behaviour. In this paper, contrary to previous works, we propose to increase the resolution from LR to HR only at the very end of the network and super-resolve HR data from LR feature maps. This eliminates the need to perform most of the SR operation in the far larger HR resolution. For this purpose, we propose a more efficient sub-pixel convolution layer to learn the upscaling operation for image and video superresolution.

The advantages of these contributions are two fold:

- In our network, upscaling is handled by the last layer of the network. This means each LR image is directly fed to the network and feature extraction occurs through nonlinear convolutions in LR space. Due to the reduced input resolution, we can effectively use a smaller filter size to integrate the same information while maintaining a given contextual area. The resolution and filter size reduction lower the computational and memory complexity substantially enough to allow super-resolution of high definition (HD) videos in realtime as shown in Sec. 3.5.
- For a network with L layers, we learn  $n_{L-1}$  upscaling filters for the  $n_{L-1}$  feature maps as opposed to one upscaling filter for the input image. In addition, not using an explicit interpolation filter means that the network implicitly learns the processing necessary for SR. Thus, the network is capable of learning a better and more complex LR to HR mapping compared to a single fixed filter upscaling at the first layer. This results in additional gains in the reconstruction accuracy of the model as shown in Sec. 3.3.2 and Sec. 3.4. We validate the proposed approach using images and videos from

publicly available benchmarks datasets and compared our performance against previous works including [7, 3, 31]. We show that the proposed model achieves state-of-art performance and is nearly an order of magnitude faster than previously published methods on images and videos.

## 2.Method

The task of SISR is to estimate a HR image  $I_{SR}$  given a LR image  $I_{LR}$  downsampled from the corresponding original HR image  $I_{HR}$ . The downsampling operation is deterministic and known: to produce  $I_{LR}$  from  $I_{HR}$ , we first convolve  $I_{HR}$  using a Gaussian filter - thus simulating the camera's point spread function - then downsample the image by a factor of  $r$ . We will refer to  $r$  as the upscaling ratio. In general, both  $I_{LR}$  and  $I_{HR}$  can have  $C$  colour channels, thus they are represented as real-valued tensors of size  $H \times W \times C$  and  $rH \times rW \times C$ , respectively. To solve the SISR problem, the SRCNN proposed in [7] recovers from an upscaled and interpolated version of  $I_{LR}$  instead of  $I_{LR}$ . To recover  $I_{SR}$ , a 3 layer convolutional network is used. In this section we propose a novel network architecture, as illustrated in Fig. 1, to avoid upscaling  $I_{LR}$  before feeding it into the network. In our architecture, we first apply a  $l$  layer convolutional neural network directly to the LR image, and then apply a sub-pixel convolution layer that upscales the LR feature maps to produce  $I_{SR}$ . For a network composed of  $L$  layers, the first  $L-1$  layers can be described as follows

$$f_l(I_{LR}; W_l, b_l) = \varphi \otimes W_l * I_{LR} + b_{l-} \quad (1)$$

$$f_l(I_{LR}; W_{1:l}, b_{1:l}) = \varphi \otimes W_l * f_{l-1} \otimes I_{LR} + b_{l-} \quad (2)$$

Where  $W_l, b_l, l \in (1, L-1)$  are learnable network weights and biases respectively.  $W_l$  is a 2D convolution tensor of size  $n_{l-1} \times n_{l-1} \times k_l \times k_l$ , where  $n_{l-1}$  is the number of features at layer  $l$ ,  $n_0 = C$ , and  $k_l$  is the filter size at layer  $l$ . The biases  $b_l$  are vectors of length  $n_l$ . The nonlinearity function (or activation function)  $\varphi$  is applied element wise and is fixed. The last layer  $f_L$  has to convert the LR feature maps to a HR image  $I_{SR}$ .

### 2.1. Deconvolution layer

The addition of a deconvolution layer is a popular choice for recovering resolution from max-pooling and other image down-sampling layers. This approach has been successfully used in visualizing layer activations [49] and for generating semantic segmentations using high level features from the network [24]. It is trivial to show that the bicubic interpolation used in SRCNN is a special case of the deconvolution layer, as suggested already in [24, 7]. The deconvolution layer proposed in [50] can be seen as multiplication of each input pixel by a filter element-wise with stride  $r$ , and sums over the resulting output windows also known as backwards

convolution [24]. However, any reduction (summing) after convolution is expensive. 1876

## 2.2. Efficient subpixel

convolution layer

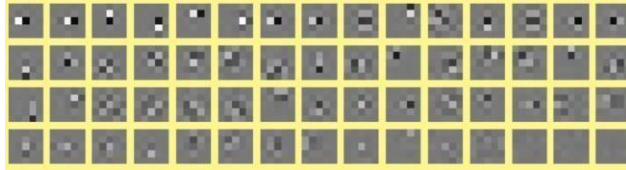


Figure 3. The first-layer filters trained on ImageNet with an upscaling factor of 3. The filters are sorted based on their variances.

The other way to upscale a LR image is convolution with fractional stride of  $r_1$  in the LR space as mentioned by [24], which can be naively implemented by interpolation, *perforate* [27] or un-pooling [49] from LR space to HR space followed by a convolution with a stride of 1 in HR space. These implementations increase the computational cost by a factor of  $r_2$ , since convolution happens in HR space. Alternatively, a convolution with stride of  $r_1$  in the LR space with a filter  $W_s$  of size  $k_s$  with weight spacing  $r_1$  would activate different parts of  $W_s$  for the convolution. The weights that fall between the pixels are simply not activated and do not need to be calculated. The number of activation patterns is exactly  $r_2$ . Each activation pattern, according to its location, has at most  $\lceil \frac{k_s}{r_1} \rceil_2$  weights activated. These

patterns are periodically activated during the convolution of the filter across the image depending on different subpixel

location:  $\text{mod}(x, r)$ ,  $\text{mod}(y, r)$  where  $x, y$  are the output pixel coordinates in HR space. In this paper, we propose an effective way to implement the above operation when  $\text{mod}(k_s, r) = 0$ :

$$\mathbf{I}^{SR} = f^L(\mathbf{I}^{LR}) = \mathcal{PS}(W_L * f^{L-1}(\mathbf{I}^{LR}) + b_L), \quad (3)$$

where PS is an periodic shuffling operator that rearranges

the elements of a  $H \times W \times C \cdot r_2$  tensor to a tensor of shape  $rH \times rW \times C$ . The effects of this operation are illustrated in Fig. 1. Mathematically, this operation can be described in the following way

$$\mathcal{PS}(T)_{x,y,c} = T_{\lfloor z/r \rfloor, \lfloor y/r \rfloor, c \cdot r - \text{mod}(y,r) + c \cdot \text{mod}(z,r)} \quad (4)$$

The convolution operator  $W_L$  thus has shape  $n_{L-1} \times r_2 C \times k_L \times k_L$ . Note that we do not apply nonlinearity to the outputs of the convolution at the last layer. It is easy to see that when  $k_L = k_s \cdot r$  and  $\text{mod}(k_s, r) = 0$  it is

equivalent to sub-pixel convolution in the LR space with the filter  $W_s$ . We will refer to our new layer as the sub-pixel convolution layer and our network as efficient sub-pixel convolutional neural network (ESPCN). This last layer produces a HR image from LR feature maps directly with one upscaling filter for each feature map as shown in Fig. 4. Given a training set consisting of HR image examples

IHR

$n, n = 1 \dots N$ , we generate the corresponding LR images  $I_{LR}^n, n = 1 \dots N$ , and calculate the pixel-wise mean squared error (MSE) of the reconstruction as an

objective function to

train the network:

$$\ell(W_{1:L}, b_{1:L}) = \frac{1}{r^2 H W} \sum_{x=1}^{rH} \sum_{y=1}^{rW} (\mathbf{I}_{x,y}^{HR} - f_{x,y}^L(\mathbf{I}^{LR}))^2 \quad (5)$$

It is noticeable that the implementation of the above periodic shuffling can be very fast compared to reduction or convolution in HR space because each operation is independent and is thus trivially parallelizable in one cycle.

Thus our proposed layer is  $\log_2 r_2$  times faster compared to deconvolution layer in the forward pass and  $r_2$  times faster compared to implementations using various forms of upscaling before convolution.

## 3. Experiments

The detailed report of quantitative evaluation including the original data including images and videos, downsampled data, super-resolved data, overall and individual scores and run-times on a K2 GPU are provided in the supplemental material.

### 3.1. Datasets

During the evaluation, we used publicly available used by SISR papers [7, 44, 3] which provides source code for multiple methods, 91 training images and two test datasets Set5 and Set14 which provides 5 and 14 images; The Berkeley segmentation dataset [26] BSD300 and BSD500 which provides 100 and 200 images for testing and the super texture dataset [5] which provides 136 texture images. For our final models, we use 50,000 randomly selected images from ImageNet [30] for the training. Following previous works, we only consider the luminance channel in YCbCr colour space in this section because humans are more sensitive to luminance changes [31]. For each upscaling factor, we train a specific network. For video experiments we use 1080p HD videos from the publicly available Xiph database which has

been used to report video SR results in previous methods [37, 23]. The database contains a collection of 8 HD videos approximately 10 seconds in length and with width and height  $1920 \times 1080$ . In addition,

we also use the Ultra Video Group database<sub>2</sub>, containing 7 videos of 1920 1080 in 1Xiph.org Video Test Media [derf's collection] <https://media.xiph.org/video/derf/> 2Ultra Video Test Sequences <http://ultravideo.cs.tut.fi/>

Figure 4. The last-layer filters trained on ImageNet with an upscaling factor of 3: (a) shows weights from SRCNN 9-5-5 model [7], (b) shows weights from ESPCN (ImageNet relu) model and (c) weights from (b) after the PS operation applied to the r2 channels. The filters are in their default ordering.



Figure 5. Super-resolution examples for "Baboon", "Comic" and "Monarch" from Set14 with an upscaling factor of 3. PSNR values are shown under each sub-figure.

size and 5 seconds in length. 3.2.

### Implementation details

For the ESPCN, we set  $l = 3$ ,  $(f_1, n_1) = (5, 64)$ ,  $(f_2, n_2) = (3, 32)$  and  $f_3 = 3$  in our evaluations. The choice of the parameter is inspired by SRCNN's 3 layer 9-5-5 model and the equations in Sec. 2.2. In the training phase,  $17r \times 17r$  pixel sub-images are extracted from the training ground truth images  $I_{HR}$ , where  $r$  is the upscaling factor.

To synthesize the low-resolution samples  $I_{LR}$ , we blur  $I_{HR}$  using a Gaussian filter and sub-sample it by the upscaling factor. The sub-images are extracted from original images with a stride of  $(17 - P_{mod}(f, 2)) \times r$  from  $I_{HR}$  and a stride of  $17 - P_{mod}(f, 2)$  from  $I_{LR}$ . This ensures that all pixels in the original image appear once and only once as the ground truth of the training data. We choose tanh instead of relu as the activation function for the final model motivated by our experimental results. The training stops after no improvement of the cost function is observed after 100 epochs. Initial learning rate is set to 0.01 and final learning

rate is set to 0.0001 and updated gradually when the improvement of the cost function is smaller than a threshold  $\mu$ . The final layer learns 10 times slower as in [7]. The training takes roughly three hours on a K2 GPU on 91 images, and seven days on images from ImageNet [30] for upscaling factor of 3. We use the PSNR as the performance metric to evaluate our models. PSNR of SRCNN and Chen's models on our extended benchmark set are calculated based on the Matlab code and models provided by [7, 3].

### 3.3. Image superresolution Results.

#### 3.3.1 Benefits of the sub-pixel convolution layer

In this section, we demonstrate the positive effect of the subpixel convolution layer as well as tanh activation function. We first evaluate the power of the sub-pixel convolution layer by comparing against SRCNN's standard 9-1-5 model [6]. Here, we follow the approach in [6], using relu as the activation function for our models in this experiment, and training a set of models with 91 images and another set with images from ImageNet. The results are shown in Tab. 1.

ESPCN with relu trained on ImageNet images achieved statistically significantly better performance compared to SRCNN models. It is noticeable that ESPCN (91) performs very similar to SRCNN (91). Training with more images using ESPCN has a far more significant impact on PSNR compared to SRCNN with similar number of parameters(+0.33 vs +0.07). To make a visual comparison between our model with the sub-pixel convolution layer and SRCNN, we visualized weights of our ESPCN (ImageNet) model against SRCNN 9-5-5 ImageNet model from [7] in Fig. 3 and Fig. 4. The weights of our first and last layer filters have a strong similarity to designed features including the log-Gabor filters [48], wavelets [20] and Haar features [42]. It is noticeable that despite each filter is independent in LR space, our independent filters is actually smooth in the HR space after PS. Compared to SRCNN's last layer filters, our final layer filters has complex patterns for different feature maps, it also has much richer and more meaningful representations. We also evaluated the effect of tanh activation function based on the above model trained on 91 images and ImageNet images. Results in Tab. 1 suggests that tanh function performs better for SISR compared to relu. The results for ImageNet images with tanh activation is shown in Tab. 2.

### **3.3.2 Comparison to the state-of-the-art**

In this section, we show ESPCN trained on ImageNet compared to results from SRCNN [7] and the TNRD [3] which is currently the best performing approach published. For simplicity, we do not show results which are known to be worse than [3]. For the interested reader, the results of other previous methods can be found in [31]. We choose to compare against the best SRCNN 9-5-5 ImageNet model in this section [7]. And for [3], results are calculated based on the  $7 \times 7$  5 stages model. Our results shown in Tab. 2 are significantly better than the SRCNN 9-5-5 ImageNet model, whilst being close to, and in some cases out-performing, the

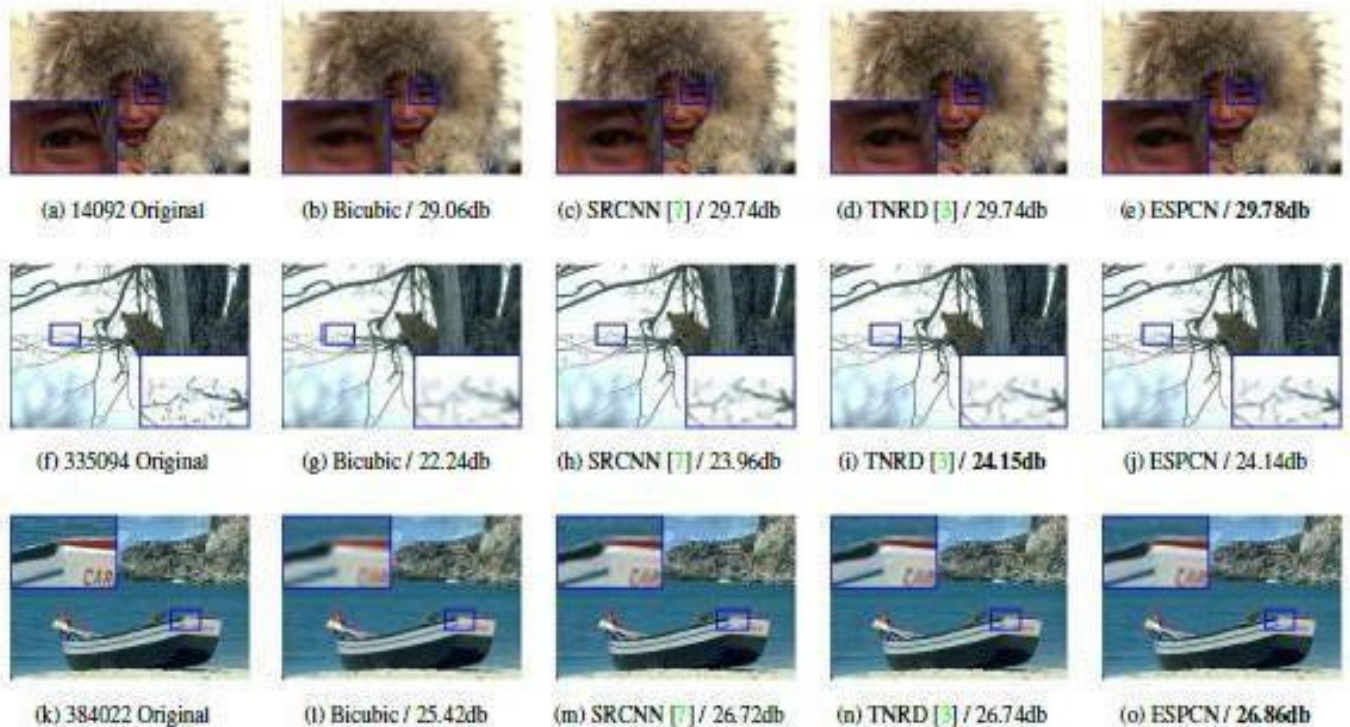


Figure 6. Super-resolution examples for "14092", "335094" and "384022" from **BSD500** with an upscaling factor of 3. PSNR values are shown under each sub-figure.

Dataset	Scale	SRCNN (91)	ESPCN (91 <i>relu</i> )	ESPCN (91)	SRCNN (ImageNet)	ESPCN (ImageNet <i>relu</i> )
Set5	3	32.39	32.39	32.55	32.52	<b>33.00</b>
Set14	3	29.00	28.97	29.08	29.14	<b>29.42</b>
BSD300	3	28.21	28.20	28.26	28.29	<b>28.52</b>
BSD500	3	28.28	28.27	28.34	28.37	<b>28.62</b>
SuperTexture	3	26.37	26.38	26.42	26.41	<b>26.69</b>
Average	3	27.76	27.76	27.82	27.83	<b>28.09</b>

Table 1. The mean PSNR (dB) for different models. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and other methods ( $p$ -value < 0.001 with paired t-test).

TNRD [3]. Although TNRD uses a single bicubic interpolation to upscale the input image to HR space, it possibly benefits from a trainable nonlinearity function. This trainable nonlinearity function is not exclusive from our network and will be interesting to explore in the future. Visual comparison of the superresolved images is given in Fig. 5 and Fig. 6, the CNN methods create a much sharper and higher contrast images, ESPCN provides noticeably improvement over SRCNN.

### 3.4. Video super resolution results

In this section, we compare the ESPCN trained models against single frame bicubic interpolation and SRCNN [7] on two popular video benchmarks. One big advantage of our network is its speed. This

makes it an ideal candidate for video SR which allows us to super-resolve the videos frame by frame. Our results shown in Tab. 3 and Tab. 4 are better than the SRCNN 9-5-5 ImageNet model. The improvement is more significant than the results on the image data, this maybe due to differences between datasets. Similar disparity can be observed in different categories of the image benchmark as Set5 vs Super Texture.

#### 3.5. Run time evaluations

In this section, we evaluated our best model's run time on Set14<sub>3</sub> with an upscale factor of 3. We evaluate the run time of other methods [2, 51, 39] from the Matlab codes provided by [40] and [31]. For methods which use convolutions including our own, a python/theano implementation is used to improve the efficiency based on the Matlab codes

provided in [7, 3]. The results are presented in Fig. 2. Our model runs a magnitude faster than the fastest methods published so far. Compared to SRCNN 9-5-5 ImageNet model, the number of convolution required to super-resolve one image is  $r \times r$  times smaller and the number of total parameters of the model is 2.5 times smaller. The total complexity of the 3It should be noted our results outperform all other algorithms in accuracy on the larger BSD datasets. However, the use of Set14 on a single CPU core is selected here in order to allow a straight-forward comparison with results from previous published results [31, 6].

super-resolution operation is thus  $2.5 \times r \times r$  times lower. We have achieved a stunning average speed of 4.7ms for super-resolving one single image from Set14 on a K2 GPU.

Utilising the amazing speed of the network, it will be interesting to explore ensemble prediction using independently trained models as discussed in [36] to achieve better SR performance in the future. We also evaluated run time of 1080 HD video superresolution using videos from the Xiph and the Ultra Video Group database. With upscale factor of 3, SRCNN 9-5-5 ImageNet model takes 0.435s per frame whilst our ESPCN model takes only 0.038s per frame. With upscale factor of 4, SRCNN 9-5-5 ImageNet model takes 0.434s per frame whilst our ESPCN model takes only 0.029s per frame.

Dataset	Scale	Bicubic	SRCNN	TNRD	ESPCN
Set5	3	30.39	32.75	<b>33.17</b>	33.13
Set14	3	27.54	29.30	29.46	<b>29.49</b>
BSD300	3	27.21	28.41	28.50	<b>28.54</b>
BSD500	3	27.26	28.48	28.59	<b>28.64</b>
SuperTexture	3	25.40	26.60	26.66	<b>26.70</b>
Average	3	26.74	27.98	28.07	<b>28.11</b>
Set5	4	28.42	30.49	30.85	<b>30.90</b>
Set14	4	26.00	27.50	27.68	<b>27.73</b>
BSD300	4	25.96	26.90	27.00	<b>27.06</b>
BSD500	4	25.97	26.92	27.00	<b>27.07</b>
SuperTexture	4	23.97	24.93	24.95	<b>25.07</b>
Average	4	25.40	26.38	26.45	<b>26.53</b>

Table 2. The mean PSNR (dB) of different methods evaluated on our extended benchmark set. Where SRCNN stands for the SRCNN 9-5-5 ImageNet model [7], TNRD stands for the Trainable Nonlinear Reaction Diffusion Model from [3] and ESPCN stands for our ImageNet model with tanh activation. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and SRCNN (p-value < 0.01 with paired ttest)

Dataset	Scale	Bicubic	SRCNN	ESPCN
SunFlower	3	41.72	43.29	<b>43.36</b>
Station2	3	36.42	38.17	<b>38.32</b>
PedestrianArea	3	37.65	39.21	<b>39.27</b>
SnowMnt	3	26.00	27.23	<b>27.20</b>
Aspen	3	32.75	<b>34.65</b>	34.61
OldTownCross	3	31.20	32.44	<b>32.53</b>
DucksTakeOff	3	26.71	27.66	<b>27.69</b>
CrowdRun	3	26.87	28.26	<b>28.39</b>
Average	3	32.41	33.86	<b>33.92</b>
SunFlower	4	38.99	40.57	<b>41.00</b>
Station2	4	34.13	35.72	<b>35.91</b>
PedestrianArea	4	35.49	36.94	<b>36.94</b>
SnowMnt	4	24.14	24.87	<b>25.13</b>
Aspen	4	30.06	31.51	<b>31.83</b>
OldTownCross	4	29.49	30.43	<b>30.54</b>
DucksTakeOff	4	24.85	25.44	<b>25.64</b>
CrowdRun	4	25.21	26.24	<b>26.40</b>
Average	4	30.30	31.47	<b>31.67</b>

Table 3. Results on HD videos from Xiph database. Where SRCNN stands for the SRCNN 9-5-5 ImageNet model [7] and ESPCN stands for our ImageNet model with tanh activation. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and SRCNN (p-value < 0.01 with paired t-test)

## 4. Conclusion

In this paper, we demonstrate that fixed filter upscaling at the first layer does not provide any extra information for SISR yet requires more computational complexity. To address the problem, we propose to perform the feature extraction stages in the LR space instead of HR space. To do that we propose a novel sub-pixel convolution layer which is capable of super-resolving LR data into HR space with very little additional computational cost compared.

Dataset	Scale	Bicubic	SRCNN	ESPCN
Bosphorus	3	39.38	41.07	<b>41.25</b>
ReadySetGo	3	34.64	37.33	<b>37.37</b>
Beauty	3	39.77	40.46	<b>40.54</b>
YachtRide	3	34.51	36.07	<b>36.18</b>
ShakeNDry	3	38.79	40.26	<b>40.47</b>
HoneyBee	3	40.97	42.66	<b>42.89</b>
Jockey	3	41.86	43.62	<b>43.73</b>
Average	3	38.56	40.21	<b>40.35</b>
Bosphorus	4	36.47	37.53	<b>38.06</b>
ReadySetGo	4	31.69	33.69	<b>34.22</b>
Beauty	4	38.79	39.48	<b>39.60</b>
YachtRide	4	32.16	33.17	<b>33.59</b>
ShakeNDry	4	35.68	36.68	<b>37.11</b>
HoneyBee	4	38.76	40.51	<b>40.87</b>
Jockey	4	39.85	41.55	<b>41.92</b>
Average	4	36.20	37.52	<b>37.91</b>

Table 4. Results on HD videos from Ultra Video Group database. Where SRCNN stands for the SRCNN 9-5-5 ImageNet model [7] and ESPCN stands for our ImageNet model with tanh activation. Best results for each category are shown in bold. There is significant difference between the PSNRs of the proposed method and SRCNN (p-value < 0.01 with paired t-test)

to a deconvolution layer [50]. Evaluation performed on an extended bench mark data set with upscaling factor of 4 shows that we have a significant speed ( $> 10\times$ ) and performance (+0.15dB on Images and +0.39dB on videos) boost compared to the previous CNN approach with more parameters [7] (5-3-3 vs 9-5-5). This makes our model the first CNN model that is capable of SR HD videos in real time on a single GPU.

## 5. Future work

A reasonable assumption when processing video information is that most of a scene's content is shared by neighbouring video frames. Exceptions to this assumption are scene changes and objects sporadically appearing or disappearing from the scene. This creates additional data implicit redundancy that can be exploited for video super resolution as has been shown in [32, 23]. Spatio-temporal networks are popular as they fully utilise the temporal information from videos for human action recognition [19, 41].

In the future, we will investigate extending our ESPCN network into a spatio-temporal network to super-resolve one frame from multiple neighbouring frames using 3D convolutions.

## 6. References

- [1] S. Borman and R. L. Stevenson, *Super-Resolution from Image Sequences - A Review*. Midwest Symposium on Circuits and Systems, pages 374–378, 1998.
- [2] H. Chang, D.-Y. Yeung, and Y. Xiong, *Super-resolution through neighbor embedding*. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 1–8. IEEE, 2004.
- [3] Y. Chen and T. Pock, *Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration*. arXiv preprint arXiv:1508.02848, 2015.
- [4] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, *Deep network cascade for image super-resolution*. In European Conference on Computer Vision (ECCV), pages 49–64. Springer, 2014.
- [5] D. Dai, R. Timofte, and L. Van Gool, *Jointly optimized regressors for image super-resolution*. In Euro graphics, volume 7, page 8, 2015.
- [6] International Conference on Machine Learning (ICML-10), pages 399–406, 2010.
- [7] B. K. Gunturk, A. U. Batur, Y. Altunbasak, M. H. Hayes, and R. M. Mersereau, *Eigenface-domain super-resolution for face recognition*. IEEE Transactions on Image Processing, 12(5):597–606, 2003.
- [8] H. He and W.-C. Siu, *Single image super-resolution using Gaussian process regression*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 449–456. IEEE, 2011.
- [9] S. Ji, M. Yang, K. Yu, and W. Xu, *3D convolutional neural networks for human action recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(1):221–231, 2013.
- [10] N. Kingsbury, *Complex wavelets for shift invariant analysis and filtering of signals*. Applied and Computational Harmonic Analysis, 10(3):234–253, 2001.
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet classification with deep convolutional neural networks*. In Advances in Neural Information Processing Systems (NIPS), pages 1097–1105, 2012.
- [12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [13] C. Liu and D. Sun, *A Bayesian approach to adaptive video super resolution*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 209–216. IEEE, 2011.
- [14] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*. arXiv preprint arXiv:1411.4038, 2014.