Multi-Frame Image Super Resolution using Convolutional Neural Networks

Individual Report

November 16, 2017

Jack Henderson, u5561978

in Collaboration with

James Russell, u5542624

ENGN8536

Abstract

We explore the application of CNNs to the problem of image super resolution, focussing on SRCNN. We replicate the implementation and confirm the performance of SRCNN, showing that simple, 3-layered, fully convolutional network is able to measurably enhance the level of detail in an image. We extend this network to a multi-frame case, which utilises the temporal relationship of subsequent video frames to add additional context to the network. Our experiments demonstrate a clear improvement of the multi-frame network over the original SRCNN. We also perform a cursory exploration of image alignment and re-introduction of colour channels with mixed results.

1 Introduction

Digital images only contain a finite amount of information, and in many cases it is often desirable to be able to extract more information out of a given image. It is a straightforward process to increase the pixel resolution of an image, for example by replacing every pixel with four identical pixels. This would result in a double of the pixel resolution of the image, but it does not add any additional information. In other words, the spatial resolution of the image is still the same. The goal of Super Resolution (SR) is to increase the spatial resolution of the image using just the information contained in the image itself. A typical super-resolution algorithm will take as input a low resolution (LR) image, and produce as output a higher resolution (HR), both in terms of pixel and spatial, image at a specified scaling factor.

The concept of Super Resolution is an inherently difficult problem to solve, as it is an example of an under-determined inverse problem. There exists a large set of similar high-resolution images that will all map to a single low resolution image. Thus, when attempting to find a mapping from a LR image to HR, there is a one-to-many relationship, and only one of these mappings represents the true high-resolution image.

Super resolution has applications in a number of different fields. Video surveillance and forensics is often an area in which increasing the level of detail in an image or video can aid in the identification of suspects, or the discovery of additional information such as licence plate numbers. The medical imaging field often uses SR techniques to combine several images taken from CT or MRI scan to create a higher resolution image. More generally, the concept of SR can be applied to any case where more detail is required in either a single image or a video.

2 Background

We provide a brief background to add context to the problem. A detailed and comprehensive literature review of super resolution concepts and techniques is presented by Yuang and Huang [1].

Interpolation is the simplest form of SR. By utilising the context of the neighbouring LR image pixels, the values for pixels in the HR image can be interpolated. A number of different interpolation methods are available, with bilinear and bicubic methods being commonly used. It follows that any arbitrary kernel can also be applied to an image to perform an interpolation, and an example of this is image de-blurring.

Another approach is to use multiple images with small levels of camera movement. This equates to a more dense sampling of the scene compared to just a single image. Statistical models are then used to fuse these images together and create a single HR image.

The recent rise of convolutional neural networks (CNNs) has prompted the exploration of a number of different CNN-based approaches to the SR problem. Fundamental to these approaches is the concept that realistic images only form a small subset of the entire space of possible images. There is a lot of structure and patterns that are present in realistic images, and this can be exploited to reduce number of potential mappings from LR to HR images, and help identify the most plausible HR image. By presenting a CNN with a large number of example images, it can learn the structures and patterns that are present and use these when performing SR on new images.

The convolutional nature of CNNs demonstrates that spatial context is key in determining the value for a particular pixel. The amount of context provided depends on the size of the convolutional kernel. Another dimension in which context can be added to the image is time. In a video sequence, the individual frames are discrete time samples from a continuous scene, similar to how adjacent pixels are discrete samples in the spatial dimensions. Thus, it follows that the temporal context provided by adjacent frames could also be useful when performing SR. This concept will form the basis of our research.

2.1 Metrics

In order to evaluate the performance of a given SR algorithm, we must have metric by which to compare them, and also a set of ground-truth LR to HR mappings which the algorithms are compared against. A common and simple metric that is used is the mean-squared error, where the error is the difference in pixel values between the ground truth HR image and the SR image produced by the algorithm. Formally, it is defined as

$$MSE = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} [I(i,j) - K(i,j)]^2$$

where m and n denote the width and height of the HR image, and I and K denote the ground-truth and SR images respectively. Another common metric used in the literature is the peak signal-to-noise ratio (PSNR), as it better represents the human perception of how different two images are.

$$PSNR = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_I^2}{MSE}\right)$$

where MAX_I^2 denotes the maximum value a pixel may take e.g. 255 for an 8-bit image.

3 Literature Review

A number of different approaches have been taken to using CNNs to perform SR. We select one of these networks, which is presented by Dong et al. [2]. Their network, SRCNN, is a simple, 3-layered, fully convolutional neural network. A graphical representation of this network is shown in Figure 1. As it is a fully convolutional network, it can be applied to any size image, although the convolution operations have no padding which makes dimensions of the output is 12 pixels smaller than the input. Adding additional padding to the input can offset this.

The input to the network is the LR image with bicubic interpolation performed to scale the image to the desired size. Dong et al. argue that a bicubic interpolation is equivalent to a de-convolution operation and thus could be added in as a preliminary layer. However due to computational optimisation and simplicity factors, the up-scaling was performed as a pre-processing step.

The network consists of 3 layers. The first is a 9 × 9 convolution kernel with 64 channels, followed by a ReLU activation function. This is described by Dong et al. as the layer responsible for "patch extraction and representation". The second layer is a 1 × 1

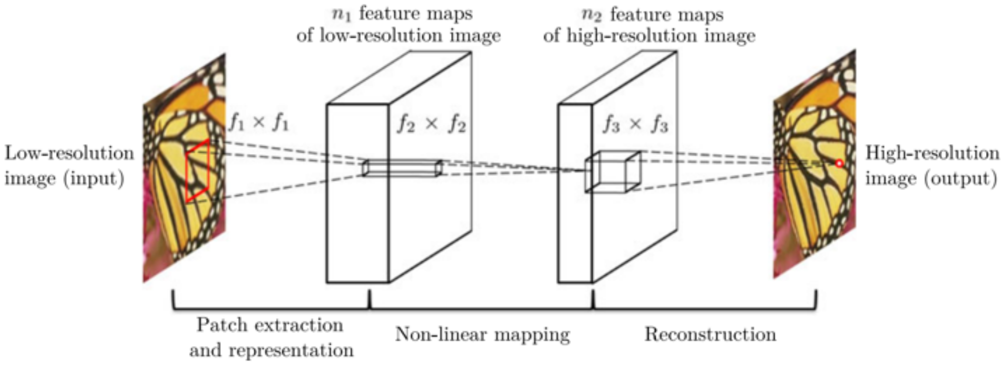


Figure 1: SRCNN network structure [2]

convolution kernel and ReLU activation with 32 channels. As it is only a 1 × 1 kernel, this layer simply provides a mapping between the 64 channel feature maps and the 32 channel maps. The addition of the ReLU introduces a non-linearity to this mapping. The final layer is a 5 × 5 convolution which outputs a single channel which is output image.

The training set used for this network consisted of 91 high-resolution images. Dong. et al. also trained their network with the ImageNet database and saw marginal improvements. Based on this, they argued that the 91 image dataset covered a sufficient range of image features. When training the net, 33 × 33 pixel patches were sampled from the set to create approximately 21,000 training samples. The HR images were used as the ground-truth, and a downsampled version was used as input to the algorithm. To simplify the network initially, colour images are transformed into the YCbCr colour space and only the Y, or luminance, channel is used. Thus, the images generated are all single channel greyscale images. They also find that introducing the remaining colour channels into the network does little to increase performance.

The network was trained using the traditional stochastic gradient descent (SGD) method with momentum. The loss function used was the sum-squared-error (SSE) of the difference between the output of the net and the original HR image.

While Dong et al. tested a number of different network structures, they found that the simple 3-layered network performed the best overall. Introducing additional layers and complexity to the network had a number of disadvantages, including increasing training time, requiring smaller learning rates, and increasing the chances of falling into a poor local minimum.

The concept of the SRCNN was used by Greaves and Winter [3] to extend the network, adding in temporal context by including adjacent frames in the input. Similar to Dong et al., they present a number of fully convolution networks and demonstrate that adding the in the temporal context can improve SR results over just the single frame case. However, the structure of the networks presented is much more complex, ranging from 5 layers up to 9 layers. While demonstrating good results, the majority of their method appeared to be a brute force trial-and-error approach of a range of different network structures in order to find the best network. This also seems to contradict the conclusions made by Dong et al. that larger network structures are not necessarily better.

This analysis of both the work from Dong et al. and Greaves and Winter gives rise to the focus of our research. Can we utilise the simple and proven structure of SRCNN and introduce a temporal context by adding adjacent frames to the input. If so, what performance benefits can we expect?

4 Methodology and Results

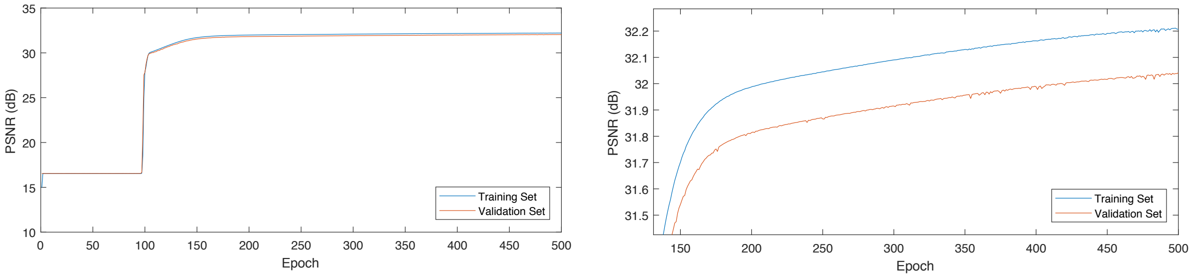
4.1 Replicating SRCNN

As we intended to extend SRCNN to a multi-frame case, our first task was to replicate the original SRCNN as described by Dong et al. The training set, and network structure, and trained network were published online[1], allowing us to inspect the code and run test examples using their pre-trained network. However, the network was implemented in the Caffe framework which we were unable to compile or install. Thus, we re-implemented the stucture, parameters and training framework of SRCNN in MatConvNet.

The training curves of our implementation are shown in Figure 2. We observed a strange phenomenon where the network failed to show any improvement in performance for approximately the first 100 epochs. This was a repeatable and consistent behaviour for which we have no explanation. After this point, we observe a sharp increase in the PSNR and then training continues as expected. It is important to note that the goal was not to replicate the performance of Dong et al.'s network, but rather to show a proof of concept that this network was able to train correctly and produce reasonable results. The amount of training performed by Dong et al. was on the order of 10^9 back-propogations, which corresponds to approximately 10^5 epochs given the size of the training set. With the limited time are hardware we had available, this was beyond the level of training that we could achieve, and thus we did not expect our network to show as high a level of performance as the pre-trained network did. We can clearly observe in Figure 2(b) that there is no sign of reaching a plateau in the training curve, and that we would see further improvement with further training.

We provide a comparison of the two networks in Figure 3. We can observe that our network does show an improvement in PSNR over the bicubic method, however it does not reach the level of performance that the pre-trained network does. This however is sufficient to give us confidence that our implementation is correct, and we believe that it is simply a matter of more training before the networks would reach the same levels of performance.

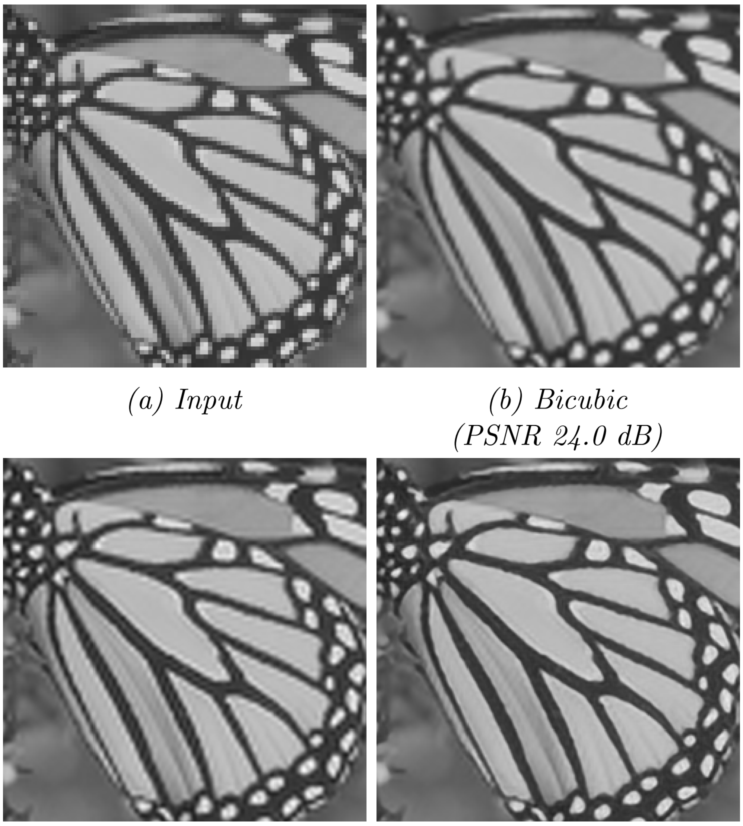
[1] Available at http://mmlab.ie.cuhk.edu.hk/projects/SRCNN.html



(a) Full Scale

(b) Zoomed

Figure 2: Training Curve of MatConvNet implementation of SRCNN



(a) Input

(b) Bicubic (PSNR 24.0 dB)

(c) MatConvNet SRCNN (PSNR 24.9 dB)

(d) Dong et al. SRCNN (PSNR 27.5 dB)

Figure 3: Comparison between Caffe and MatConvNet Implementations

4.2 Multi-Frame SRCNN

As we were satisfied that our implementation of SRCNN was equivalent to the reference implementation, we examined the multi-frame case. One of the key observations that Greaves and Winter make is that adding too much temporal context can negatively affect results as the scene is changing over time and subsequent frames carry increasingly less information about the frame of interest. Thus we restrict our temporal context to only the frame immediately before and after the frame of interest. These additional frames undergo the same pre-processing so that they are the same size as the frame of interest. This changes the dimensions of the input from $m \times n \times 1$ to $m \times n \times 3$

This presented the challenge of deciding how to modify the network to account for the additional input. Based on the description of the functions of each layer that Dong et al. propose, we theorise that it is only necessary to change the first layer. This layer is responsible for "patch extraction and representation", which is where adding the additional temporal context will affect the network. Once this information is encoded in the first layer, the functions of the second and third layer do not change significantly, as they still perform a non-linear mapping from low- to high-resolution feature maps, and then the image reconstruction. Based on this, we modified the kernel of the first layer from a $9 \times 9 \times 1$ convolutional kernel to a $9 \times 9 \times 3$ which matches the change in input from an $m \times n \times 1$ to an $m \times n \times 3$.

4.2.1 Initialisation

While it would be possible to train this new network from a random initialisation of weights, we consider a more refined approach. As we are only modifying the first layer of the network, the overall structure remains the same as the single frame network. If we consider the case where the three frames of the input are identical, we would expect the same output as what the single frame network would provide, as the additional frames are providing no additional information. Thus, we suggest that a sensible initialisation point for the weights of the network would be to use the weights from a pre-trained single frame SRCNN. As the kernel in the first layer has changed dimension, we stack 3 copies of the $m \times n$ kernels to create an $m \times n \times 3$ kernel and multiply this kernel by 1/3 to normalise.

Recalling that we did not fully train out network to the level described by Dong et al., we instead use the pre-trained network that they provide to initialise our multi-frame network. However, even with this refined initialisation, a significant amount of training must still be performed, as the network must learn how to utilise the minor differences between subsequent frames in order to further increase the level of detail in the image.

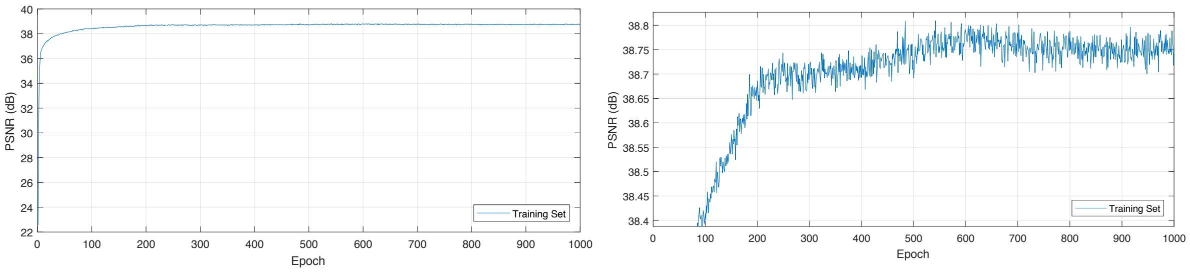
4.2.2 Training Set

Another challenge of this project was to identify an appropriate dataset to train the network on. Each sample consists of 3 consecutive frames from a video sequences, however we could not simply use a large number of frames from a single video sequence, as this would not provide enough variation in image features to train the network effectively. This we restricted the dataset to only one sample from any given video sequence. We were also not able to find an appropriate dataset available online, and Greaves and Winter do not make their dataset available. We instead combine a number of samples from both the 2016 Visual Object Tracking Benchmark dataset [4], and a dataset referred to by Greaves and Winter [5]. From these sources we obtain 21 samples from unique video sequences, and while this is a smaller number than the single frame case, many of these videos were of a higher resolution, and so after sampling patches from the dataset, a similarly sized training set of over 20,000 samples resulted.

4.2.3 Training and Results

Once initialised with the pre-trained weights, we trained the multi-frame SRCNN network for 1000 epochs. The training curves are shown in Figure 4. By initialising the weights with the pre-trained network, we avoid the peculiarities observed in the single frame network and performance is reasonable from the beginning. We also observe that the network reaches a much higher PSNR value of over 38 dB compared to the single frame case of only 31-32 dB. This provides an initial indication that the multi-frame temporal context is enabling better performance from the network. We also observe that the training curve appears to be more 'noisy' in that the value fluctuates more than in the single frame case. This is potentially due to the increase in the number of parameters from $\approx 8,100$ to $18,000$. If this is the case, then decreasing the learning rate would reduce these fluctuations, but also lengthen the training time.

In Figure 5, we compare the results from the single frame and multi-frame case. In both cases the networks have not been trained to completion due to resource constraints and thus the results are not representative of the maximum achievable performance. However, they do allow for a comparison between the two networks as they were trained



(a) Full Scale

(b) Zoomed

Figure 4: Training Curve of MatConvNet implementation of SRCNN

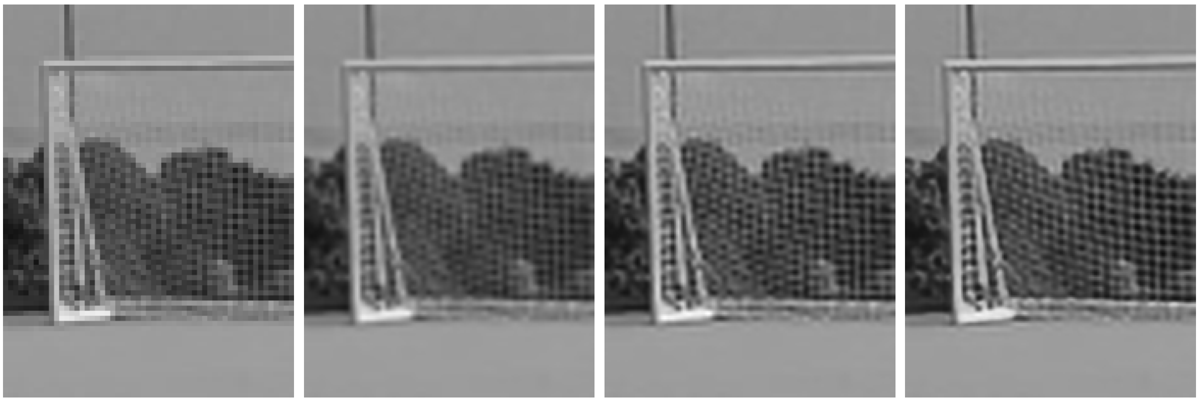
for similar amounts of time. The ultimate goal is to demonstrate a proof-of-concept that this technique is worthwhile pursuing further.

As can be seen in Figure 5, the single frame network is able to provide an improvement over the bicubic interpolation method, which we use as a baseline measure of performance. Further to this, the multi-frame network provides an additional increase in image quality. This is most apparent in the netting of the goal, which contains a lot of high-frequency information. In this video, the camera is hand-held and moves slightly from frame to frame. This allows a slightly different view of the scene to be captured by each frame, and this information can be combined to resolve more detail than is available in just a single frame.

While we generally observe a consistent performance improvement from the multi-frame network over the single-frame network, there are some cases where the results are not as good. Figure 6 shows one such example of a basketball match where both the single-frame and multi-frame perform worse than the bicubic interpolation. What is more concerning is that the multi-frame network performs worse than the single-frame network, showing that adding the additional context does not always improve results.

4.3 Investigating Image Alignment

Fundamental to the process of the multi-frame network is the assumption that the surrounding frames represent approximately the same view of the scene. However, if the camera is not fixed, then the view represented in the image is continuously changing. In other words, a single pixel location in two different frames does not represent the same point in space. We notice that in some of our test cases, there is a significant amount of camera movement. For example, in the basketball scene shown in Figure 6, the camera is panning across the court to track the player. This could potentially be one of the reasons



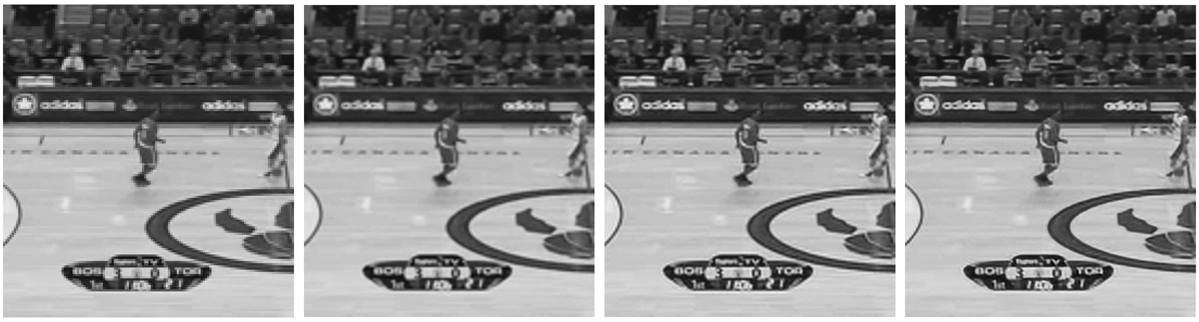
(a) Input

(b) Bicubic (PSNR 24.2 dB)

(c) SF-SRCNN (PSNR 25.0 dB)

(d) MF-SRCNN (PSNR 26.0 dB)

Figure 5: Example of Multi-Frame SRCNN vs Single Frame



(a) Input

(b) Bicubic (PSNR 23.3 dB)

(c) SF-SRCNN (PSNR 23.0 dB)

(d) MF-SRCNN (PSNR 22.7 dB)

Figure 6: Example of poor performance from both Single- and Multi-Frame SRCNN

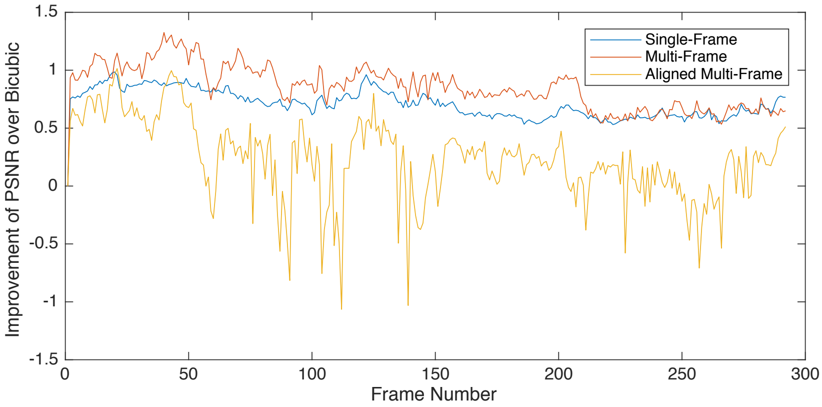


Figure 7: Results of each method on 'Bolt2' video sequence compared to Bicubic Interpolation (best viewed in colour)

why the performance is not as good as other sequences with less movement.

We decided to investigate whether performing an image alignment in the pre-processing step would provide better performance for the multi-frame SRCNN. Using MATLAB's inbuilt image registration toolbox, we implemented an image registration function which aligned the two contextual frames to the frame of interest. The function utilised SURF points to generate a projective transformation which was then applied to the images.

A preliminary test was performed on a video of an athletics race, with the results shown in Figure 7. There is a high amount of camera movement in this video as the camera pans down the race track to follow the athletes. Consequently, we can observe that the multi-frame network provide little to no improvement over single-frame network, and it makes this an ideal case to test whether image alignment has any benefits. However, surprisingly, when the image alignment was performed we found that results were significantly worse than the non-aligned multi-frame network, and often at times worse than the baseline bicubic method.

There are a number of possible explanations for the drop in performance. One is that there were several artefacts in the SR image around the border. This was due to the image alignment process as the transformed image no longer fits directly into the same pixel space as the frame of interest. For example, if the camera pans from left right, aligning the images will cause a black border on the right side of the frame preceding the FOI. We also observed that the transformations were not always consistent, and there was some variability in the quality of the image alignment.

Another example of the comparative performances is shown in figure 8, which shows

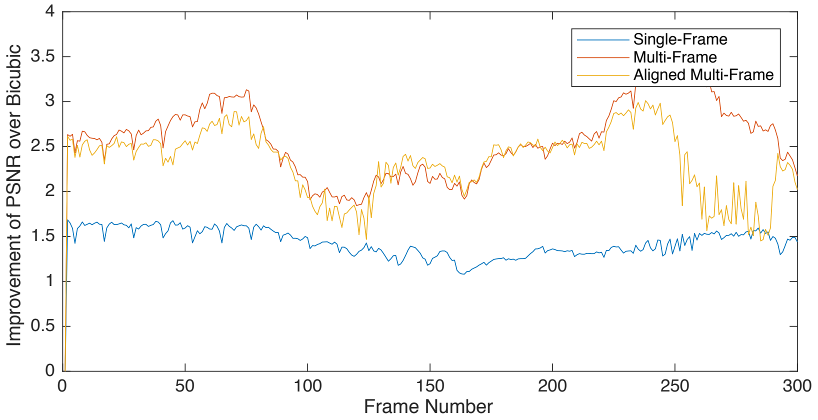


Figure 8: Results of each method on 'Basketball' video sequence compared to Bicubic Interpolation (best viewed in colour)

that the multi-frame network is consistently better performing than the single-frame case. Again, the image alignment pre-processing fails to show any improvement. The section of the video between frames 100 and 200 is where the camera is panning very quickly across the scene, and we would expect that, if there was any improvement to be made, this is a prime case where image alignment should improve performance.

While these results are not positive, we still believe that image alignment can have a positive effect on performance, and that further investigation is required. However, what is clear is that the process is not as straightforward as initially expected.

4.4 Adding colour to images

As discussed earlier, the SRCNN only utilises the Y, or luminance, channel of the YCbCr colour space. This is due to the fact that human eyes are more sensitive to changes in luminance than they are to colour [6]. Dong et al. also experiment with adding the chroma channels as additional input and also with using the RGB colour space, but found little advantage in doing do.

A common technique in video compression is to use chroma sub-sampling, where the luminance channel is stored at full resolution, and the two chroma channels, Cb and Cr, and sampled at a lower spatial resolution. For example, in 4:2:0 sub-sampling, the chroma channels are sampled at a quarter of the rate of the luminance channel. The effect of this is barely perceivable by humans as we are more sensitive to changes in the luminance channel compared to the chroma channels.

We can utilise a similar approach to reconstruct super resolution colour images from a low resolution colour image. Firstly, we extract the Y channel and feed it through the network to create a SR image of the Y channel. Then, a simple upsampling method, like bicubic upsampling, is performed to the Cb and Cr channels, which are then recombined with the SR Y channel to create a full colour SR image. An example of this process is shown in Figure 9.

5 Future Work

There are a significant number of areas open to further investigation and exploration. Immediate next steps to this research include performing more training on the network, and increasing the size of the multi-frame training set. We acknowledge that the performance of the networks shown in this paper doesn't realise the fully potential of what they are capable of, and further training with a larger dataset would help to achieve this. Further to this, experimentation with the structure of the network could prove valuable, especially in terms of increasing the number of channels in each layer.

One of the key areas that should be investigated further is the image registration process. Our investigation into this was relatively cursory, and we believe that better performance is achievable with the right image registration techniques. Simplifying the transformation from a projective transform to an affine, or even translational transformation could be a viable approach.

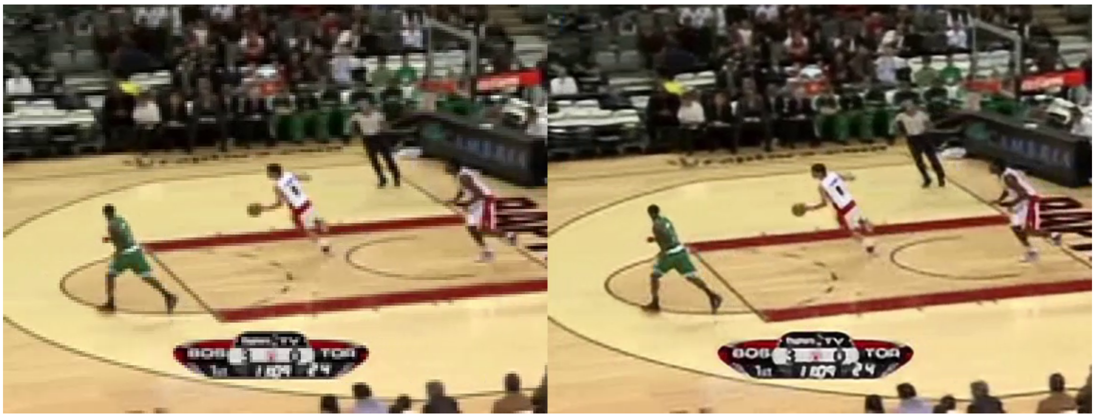


Figure 9: Example of coversion from LR colour image (left) to SR colour image (right) by upsampling chroma channels

6 Conclusion

In this project we explored the application of convolutional neural networks to the image super resolution problem. We successfully replicate a state-of-the-art network, SRCNN, in Matlab and demonstrate an improvement over the baseline level of performance. Our key contribution was to extend this network to include temporal context by adding additional video frames to the input and modifying the network to suit. We also collate a multi-frame training set in order to train the MF-SRCNN. Our testing demonstrates that the addition of temporal context can increase performance in most cases. Further, we explore the effects of image alignment, and find that performance was not improved over the standard multi-frame network. Finally, we demonstrate the addition of colour to the SR image, and also the application to full-length video sequences.

References

[1] J. Yang and T. Huang, "Image super-resolution: Historical overview and future challenges," 2010.

[2] C. Dong, C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[3] A. Greaves and H. Winter, "Multi-frame video super-resolution using convolutional neural networks," 2016.

[4] M. Kristan, A. Leonardis, J. Matas, and et.al, "The visual object tracking vot2016 challenge results," in *14th European Conference on Computer Vision*, 2016.

[5] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 531–539. DOI: `10.1109/ICCV.2015.68`.

[6] S. Winkler, C. J. van den Branden Lambrecht, and M. Kunt, "Vision and video: Models and applications," *Vision models and applications to image and video processing*, 2001.