18 Jun 2021

Single-Image Super Resolution using Convolutional Neural Network

William Symolon

Cihan H. Dagli
Missouri University of Science and Technology, dagli@mst.edu

Recommended Citation

W. Symolon and C. H. Dagli, "Single-Image Super Resolution using Convolutional Neural Network," *Procedia Computer Science*, vol. 185, pp. 213-222, Elsevier B. V., Jun 2021.
The definitive version is available at https://doi.org/10.1016/j.procs.2021.05.022

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 185 (2021) 213–222

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference Theme: Big Data, IoT, and AI for a Smarter Future
Malvern, Pennsylvania, June 16-18, 2021

Single-Image Super Resolution Using Convolutional Neural Network

William Symolon* and Cihan Dagli

Engineering Management and Systems Engineering Department
Missouri University of Science and Technology, Rolla, MO 65409

Abstract

Increasing threats to U.S. national security satellite constellations have resulted in an increased interest in constellation resilience and satellite redundancy. CubeSats have contributed to commercial, scientific and government applications in remote sensing, communications, navigation and research and have the potential to enhance satellite constellation resilience. However, the inherent size, weight and power limitations of CubeSats enforce constraints on imaging hardware; the small lenses and short focal lengths result in imagery with low spatial resolution. Low resolution limits the utility of CubeSat images for military planning purposes and national intelligence applications. This paper implements a super-resolution deep learning architecture and proposes potential applications to CubeSat imagery.

Keywords: super resolution; CNN; CubeSats

1. Introduction

CubeSats [5] have the demonstrated potential to contribute to commercial, scientific and government applications in remote sensing, communications, navigation and research at a fraction of the size, development costs and launch costs of the large, multi-function satellites designed to support Cold War military requirements. However, the reduced size, weight and power margins inherent in CubeSats also have disadvantages. Smaller satellites are typically limited to single payloads or functions. For traditional electro-optical (EO) imagery applications, high resolution (HR) requires large lenses and long focal lengths, which in turn require large satellites to support them [3]. Past research has demonstrated that on-board image processing techniques can make more efficient use of limited satellite resources

10.1016/j.procs.2021.05.022

[4, 6, 9, 13]. Work in pixel registration [13], feature classification [4], parallel computing [6], and radar interferometry [7] has laid the groundwork for the collection of EO imagery using multiple CubeSats flying in close formation.

Despite the data handling improvements, there remains one fundamental limitation of CubeSats for EO imaging applications: the small lenses and short focal lengths result in imagery with low spatial resolution. These low resolutions (LR) are sufficient for scientific applications such as weather forecasting and agricultural assessments [14, 17], but are insufficient for defense mission planning and intelligence operations. There are two primary methods for improving spatial image resolution: hardware solutions focus on improved camera capabilities and analytical methods that focus on software solutions [9]. Hardware improvements are often restricted by cost, large size, or technology readiness limitations – all three of which are impractical for the CubeSat concept. Additionally, optical imaging hardware is subject to the Rayleigh criterion in which light diffraction limits the best possible resolution [11]. Thus, a computational algorithm solution is required to improve EO spatial resolution of CubeSat images.

This paper proposes a deep learning pipeline using a Convolutional Neural Network (CNN) to implement a single-image super resolution (SISR) model. Section 2 provides background context for the super-resolution problem. Section 3 discusses the development of the CNN model and selection of the model performance metrics. Section 4 covers model performance for various hyperparameters, in terms of the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM), as compared with a benchmark single-image super-resolution (SISR) model, the Efficient Sub-Pixel Convolutional Neural Network (ESPCN) [18]. Section 5 discusses areas of future work, followed by concluding remarks in section 6.

2. Background

There are two primary methods of image super-resolution: single-image super-resolution (SISR) and multi-image super-resolution (MISR) [15]. SISR requires a training database of low-resolution (LR) and high-resolution (HR) pairs with specific features and segments common to both and annotated for machine learning algorithms. There are three main categories of SISR algorithms: interpolation-based algorithms reconstruct HR images using existing pixels to interpolate probable missing pixels; reconstruction-based algorithms use *a priori* knowledge (down-sampling, blurring and warping) to recover the HR image; learning-based algorithms use dictionary pairs of training and testing images to estimate HR images [21]. SISR techniques are fast, less computationally intensive and are capable of producing sharp HR images for specific applications.

Multi-image super-resolution (MISR) is a well-studied problem which typically consists of three stages: registration estimates the shifts between LR images, relative to a reference image, with sub-pixel accuracy; interpolation obtains a uniform HR image from a non-uniform composite of LR images; and restoration removes the image blur and noise. MISR can be further sub-divided into frequency domain techniques and spatial domain techniques [19]. The relative motion between LR input images produces the sub-pixel shifts necessary to achieving higher resolution enhancement by accounting for information from adjacent image frames, given sufficiently accurate motion estimation [2].

Shi, et al describe some of the underlying research that has gone into their selection of a CNN to perform the SISR operation [18]. A CNN is trained to recognize specific patterns in an LR image and represent those patterns in an HR image. The convolutional layers allow the training of pattern recognition to be less computationally complex compared to more traditional approaches such as decision forests and backpropagation. CNNs do not synthesize a high-resolution image from low resolution images, which is the goal of MISR techniques.

3. CNN Model Development and Training

The CNN model proposed in this paper is based on the Efficient Sub-Pixel Convolutional Neural Network (ESPCN) introduced by Shi, et al in 2016 [18]. The model uses a series of convolution layers to develop low-resolution feature maps which are then converted to a high-resolution image at the final layer. The equations governing the behavior of the input and hidden layers are those developed by Shi, et al and are shown in equations (1) and (2) below [18].

$$f^1(I^{LR};\, W_1, b_1) = \phi(W_1 * I^{LR} + b_1) \tag{1}$$

$$f^L(I^{LR};\, W_{1:L}, b_{1:L}) = \phi(W_L * f^{L-1} I^{LR} + b_L) \tag{2}$$

Where L is the number of network layers, I^{LR} is the LR input image, and $W_L, b_L, l \in (1, L-1)$ are learnable network weights and biases respectively. W_L is a 2D convolution tensor of size $n_{L-1} \times n_L \times k_L \times k_L$, where n_L is the number of features at layer L, n_0 is the number of bands in the image, and k_L is the filter size at layer L.

Finally, the output layer performs the sub-pixel convolution to output the super-resolved image. Again, the output layer of the model uses the equations developed by Shi, et al, shown in equation (3) below [18].

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

Where I^{SR} is the super-resolved output image and PS is a periodic shuffling operator that reshapes tensor elements from $H \times W \times C \cdot r^2$ to $rH \times rW \times C$, via the mathematical representation shown in equation (4). Note that the authors were able to avoid this periodic shuffling during network training and instead pre-shuffled the training data to match the required output, thus saving computational time during network training.

$$PS(T)_{x,y,c} = T_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, C \cdot r \cdot mod(y,r) + C \cdot r \cdot mod(x,r) + c} \tag{4}$$

A graphical representation of this shuffling operation can be seen in Figure 1 below.

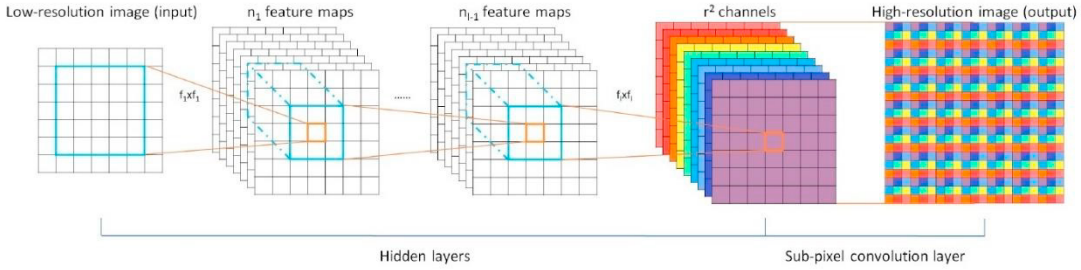

Fig. 1. Diagram of the Efficient Sub-Pixel Convolutional Neural Network developed by Shi, et al. [18]

With this ESPCN model as a basis for reference, this paper uses a similar CNN as shown in Figure 2 below, adapted from [12]. Note that while the ESPCN model illustrated in Figure 1 uses two hidden layers before the sub-pixel convolution output layer, the model used in this paper includes two additional hidden layers to extract additional feature maps, as well as a dropout regularization layer, at the cost of a slightly increased training time.

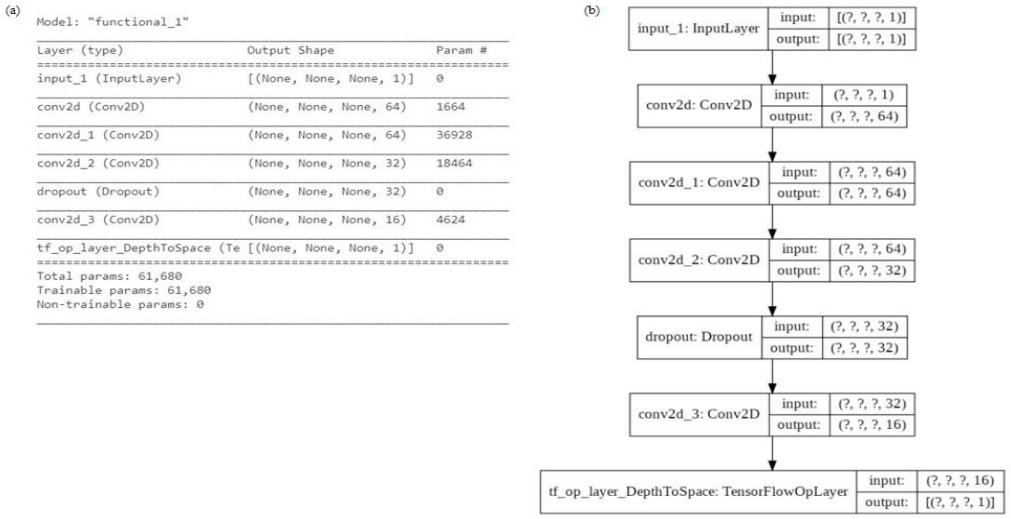

Fig. 2. (a) Python-generated summary of the CNN model implementation; (b) Python plot representation of the same CNN network [11]

One additional modification made to the original ESPCN network is the conversion of the input images from the Red, Green, Blue (RGB) color space to the Luminance, Chrominance (YCbCr) color space. This is a lossless, reversible transformation that results in a single luminance signal (Y), which is a weighted combination of the R, G, and B color bands. The transformation also results in two chrominance or color difference signals, a blue projection (Cb) and a red projection (Cr). The Y signal is then split out of the resulting transformation and passed as a single channel image input to the CNN. Using a single channel input image has the advantage of reducing the computational complexity required of the CNN. The resulting enhanced output Y channel is then recombined with the original Cb and Cr channels and transformed back to the RGB color space for the final super-resolved image.

This color space transformation method is effective because human visual perception is much more sensitive to variations in brightness (luminance) than chrominance [10]. Thus, the network achieves an improvement in the apparent resolution.

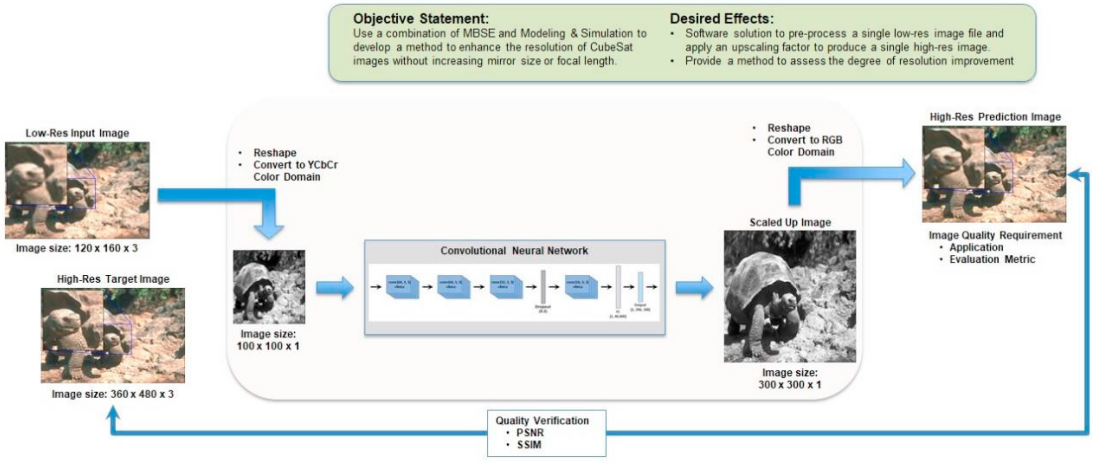

Fig. 3. OV-1 for Single Image Super Resolution CNN implementation

Any image processing algorithm requires an objective measure of performance, typically via quantitative image quality assessment methods, as described by Sara, et al [16]. This model uses two common measures of performance, Peak Signal to Noise Ratio (PSNR) [8] and the Structural Similarity Index (SSIM) [20]. PSNR is the ratio between the maximum signal and the corrupting noise that affects the high-resolution reconstruction (i.e. the

super-resolved image output by the CNN). PSNR is typically expressed in terms of a logarithmic decibel scale where a larger number indicates better performance, as determined by equation (5).

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$
$$= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10} MSE \quad (5)$$

Where MAX_I is the maximum possible pixel value of the image; in the case of an 8-bit color scale, $MAX_I = 255$.

While PSNR is simple to understand and implement and has a clear physical meaning, it often doesn't cleanly align with perceived image quality [16]. Therefore, this model also uses as a normalized reference method to predict the perceived quality of digital images, using the similarity between two images [16, 20]. The SSIM method is predicated on the assumption that spatially close pixels have strong inter-dependencies. That is, pixels that are spatially close are more likely to represent the same object within an image. SSIM is expressed as a decimal between 0 and 1, where 1 indicates perfect similarity. SSIM is determined by equation (6).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (6)$$

Where μ_x and μ_y are the averages of x and y, σ_x^2 and σ_y^2 are the variances of x and y, σ_{xy} is the covariance of (x, y), and c_1 and c_2 are stabilization terms in the case of very small denominator values. $c_1 = (k_1 L)^2$ and $c_2 = (k_2 L)^2$, where L is the dynamic range of the pixel values and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

This model uses both PSNR and SSIM to evaluate the quality of the output images, as discussed in the following section.

4. Model Results

This section compares the results of the CNN described above against the original ESPCN described by Shi, et al [18]. The CNN was trained with a variety of learning rates, optimizers, and scaling factors as well as with and without a 30% dropout regularization. Table 1 shows the hyperparameters used during the model training process. All model configurations were trained using 100 epochs and a default scaling factor of three (i.e. the model attempts to generate a super-resolved image with three times the resolution of the input image).

Table 1: CNN hyperparameter variations

Learning Rate	Optimizer	Dropout Regularization
0.01	Adam	30%
0.001	RMSprop	None
0.0001	Adagrad	

The model was trained using the University of California, Berkeley, Computer Vision Group, Berkeley Segmentation Data Set 500 (BSDS500) [1], consisting of 500 training images and 200 testing images with an 80/20 training/validation split.

Figure 4 shows a sample of the resulting super-resolved images from the various hyperparameter combinations in table 1. The Adam optimizer with a learning rate of 0.001 and dropout regularization of 30% showed a slightly better performance than with no dropout. More experimentation would be necessary to know whether or not that configuration yields consistently better performance or if the improvement was due to the stochastic nature of the deep learning pipeline.

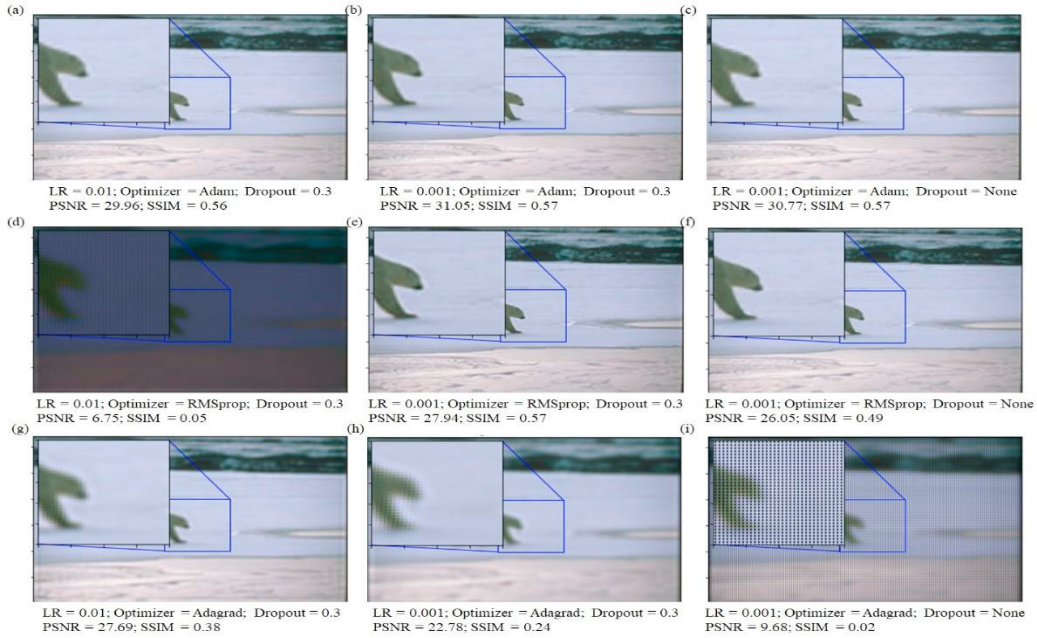

Fig. 4: Results of CNN hyperparameter training variations

Having identified the CNN hyperparameter configuration with the best performance, the next step is to experiment with changing other factors in the training scenario. The additional modifications included increasing the number of training epochs and experimenting with different scaling factors, as shown in Table 2.

Table 2: Additional CNN hyperparameter training variations

Fixed Hyperparameters	Scaling Factor	Epochs
LR = 0.001	2	100
Optimizer = Adam	4	200
Dropout = 30%		

Figure 5 shows the model training results for varying image scaling factors and training epochs; note that figures 4(b) and 5(b) were generated from the same model configuration. As expected, changing the scaling factor of the model has a significant effect on the super-resolved image output by the model. Changing the scaling factor is analogous to zooming in or out on an image; a larger scaling factor has the effect of zooming in and causing additional pixilation.

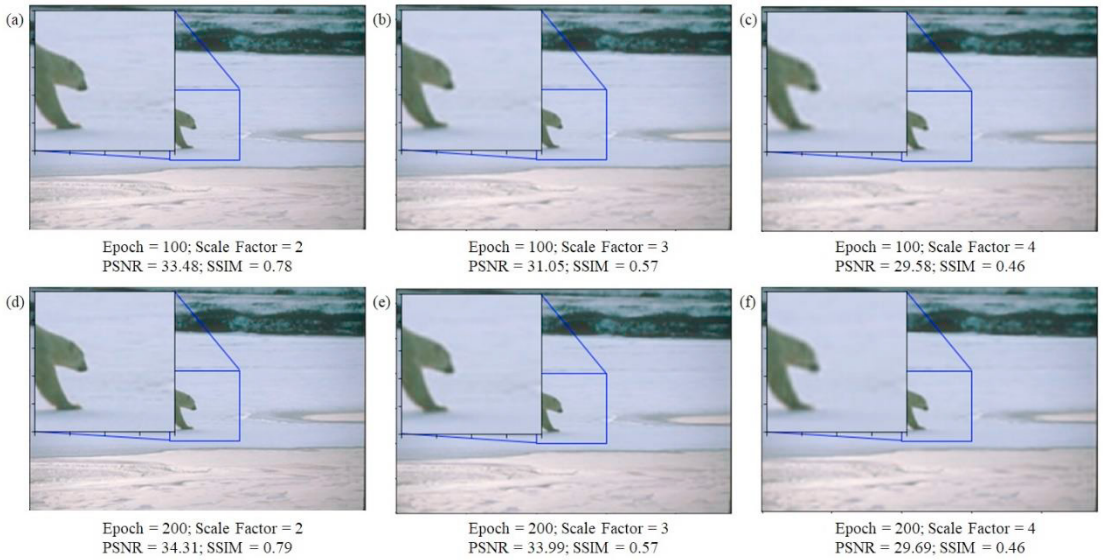

Fig. 5:Training results from CNN epoch number and scaling factor training variations

Interestingly, while increasing the number of training epochs does improve the model performance, the improvement is slight (0.83 dB) for a scaling factor of two, moderate (2.94 dB) for a scaling factor of three and negligible (0.11 dB) for a scaling factor of four. This may be due in part to the fact that Shi, et al originally designed their ESPCN model to scale images by a factor of three [18] and in part due to the stochastic nature of deep learning. Depending on the desired application for the super-resolved images, the user can decide whether or not the slight performance improvement from additional training epochs justifies the extended training time.

As an example of the subjective visual comparison of the resolution improvement, Figure 6 shows a side-by-side comparison of the low-resolution input, high-resolution target, and super-resolved model output. Visual inspection of Figure 6 shows that the model is able to produce an image that has a resolution better than the input image but not yet approaching the resolution of the target image.

The final step in performance analysis is to evaluate a selection of the MSE plots generated during the model training process. Recall that PSNR is determined by the maximum pixel value and the MSE (equation (5)). Figure 7 confirms that the Adam optimizer does tend to outperform RMSprop and Adagrad, regardless of the other hyperparameter settings.

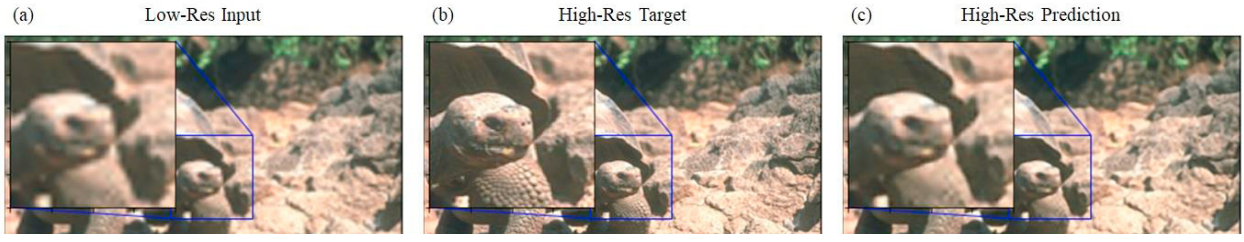

Fig. 6: Side-by-side comparison of (a) low-res input image, (b) high-res target image, and (c) super-resolved model output

In their paper proposing the ESPCN network, Shi, et al compared the performance of their network against a selection of publicly available benchmark datasets, including the Berkeley segmentation dataset BSDS500 [1]. In all cases, Shi, et al were able to achieve superior performance in terms of PSNR and model run time. This paper only uses BSDS500 dataset however, given the extensive comparisons across multiple datasets by Shi, et al, comparing the performance of this CNN with the ESPCN will provide a snapshot of capabilities, in terms of PSNR. Shi, et al also use model run time of the ESPCN compared to other super resolution techniques and networks such as bicubic, SRCNN and TNRD [18]. Shi, et al conducted their experiments using a K2 GPU which was not available for training this model. Since it isn't possible to conduct a viable comparison of performance time without comparable hardware, the performance comparison is limited to PSNR without consideration of the time required to train the network or to super-resolve a single image. Shi, et al did not report performance results using the SSIM metric. Table 3 shows a comparison of the average PSNR scores between the CNN in this paper and the ESPCN.

In an effort to make the performance comparison as relevant as possible, the model hyperparameters were set to equal those used by Shi, et al in the ESPCN network. The learning rate was set to 0.0001 and the network was trained for 100 epochs. The comparison varied the optimizer, whether or not the network uses dropout regularization, and the scaling factor between 3 and 4 (Shi, et al did not report results using a scaling factor of 2).

Table 3 clearly shows that the original ESPCN network proposed by Shi, et al outperforms the CNN modifications made for this paper using the BSDS500 images. Shi, et al did not report which optimizer they used to train their network, nor did they report whether or not they used any type of regularization.

A final consideration when evaluating network performance differences is the fact that this CNN only enhanced a single band (Y) of the input image, whereas Shi, et al enhanced all three bands of the low-resolution RGB input image. The inclusion of the two additional hidden layers and the dropout regularization helped improve the

Table 3: Comparison of average PSNR with baseline ESPCN network

ESPCN Scaling Factor 3 Average PSNR = 28.64			**ESPCN Scaling Factor 4** Average PSNR = 27.07		
Optimizer	**Dropout**	**Avg PSNR**	**Optimizer**	**Dropout**	**Avg PSNR**
Adam	30%	26.55	Adam	30%	25.52
	None	26.79		None	25.78
RMSprop	30%	25.28	RMSprop	30%	25.21
	None	25.87		None	24.76
Adagrad	30%	17.71	Adagrad	30%	14.82
	None	14.49		None	12.63

performance of this network, such that it begins to approach the benchmark ESPCN performance established by Shi, et al. However, that improvement does not offset the loss of performance incurred from enhancing only one color band versus enhancing all three bands. Additional information regarding nuances of the network and details of the training methodology would enable a more comprehensive comparison of network performance.

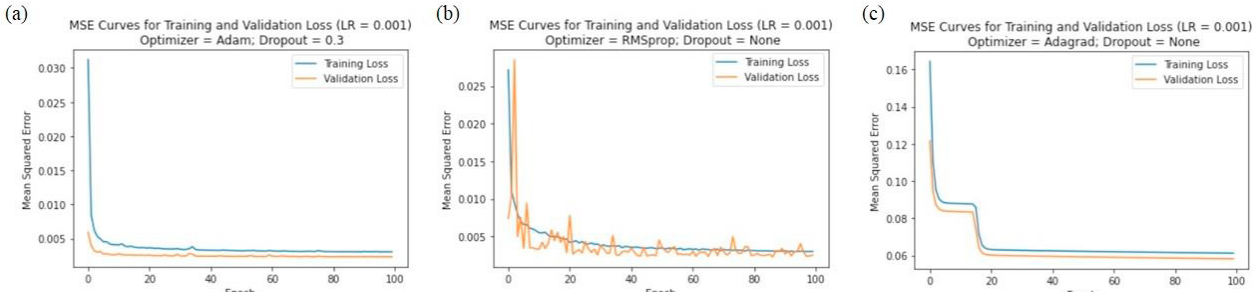

Fig. 7: Selected MSE plots from CNN hyperparameter training variations

5. Future Work

The original concept for this model was to implement a multi-image super resolution deep learning pipeline capable of receiving four low-resolution images and outputting a single high-resolution image. Project timelines and implementation difficulties resulted in the need for a simplified single-image super resolution network architecture. Future work will expand the model to be capable of accepting multiple low-resolution image inputs and synthesizing a single high-resolution image.

As noted in section 4, the CNN network used in this paper only enhanced a single band (Y) of the input image, resulting in reduced computational complexity which then also caused a corresponding reduction in network performance. Developing a network capable of improving resolution across all three color bands is an important area of future study.

Since the ultimate goal of the research is to improve CubeSat image resolution, the network will need to be modified to accept low-resolution, overhead, remote sensing images which generally have a high number of very small objects within the image frame. Future versions of the network must be sensitive enough and have a sufficient number of layers to extract a large number of feature maps and synthesize those feature maps into a coherent high-resolution image.

As a final goal for the research, in addition to being capable of accepting multi-image inputs, the network would be dynamically scalable, capable of identifying the number of low-resolution input images being provided and generating a network of sufficient breadth and depth to process those images.

6. Conclusion

Traditional high-resolution, electro-optical satellite imagery requires large lenses and long focal lengths, which in turn require large satellites to support them [3]. Of the two primary methods for improving spatial image resolution, hardware solutions are less practical for CubeSats due to the larger costs, physical size, and diffraction-limited resolution commonly associated with current state of the art imaging hardware. Thus, a software solution is an attractive alternative to improve EO spatial resolution of CubeSat images.

This paper implements a modified version of the ESPCN proposed by Shi, et al [18] and compares network performance in terms of the super-resolved image PSNR. The initial results indicate that this network does not achieve an average PSNR as high as that of the ESPCN. Nevertheless, the PSNR and SSIM results obtained from this network still provide a measurable improvement in resolution over the low-resolution input images and represent a step toward providing usable high-resolution imagery from low-resolution CubeSat images.

References

[1] Arbelaez, Pablo, Maire, Michael, Fowlkes, Charless, and Malik, Jitendra. (2011) “Contour Detection and Hierarchical Image Segmentation.” *IEEE Transactions on Pattern Analysis for Machine Intelligence,* **33(5)**: 898–916. http://dx.doi.org/10.1109/TPAMI.2010.161.

[2] Bätz, Michel, Eichenseer, Andrea, Seiler, Jürgen, Jonscher, Markus, and Kaup, André. (2015) “Hybrid Super-Resolution Combining Example-Based Single-Image and Interpolation-Based Multi-Image Reconstruction Approaches.” *Proceedings – International Conference on Image Processing, ICIP*: 58–62.

[3] Buzzi, Pau Garcia, Selva, Daniel, Hitomi, Nozomi, and Blackwell, William J. (2019) “Assessment of constellation designs for earth observation: Application to the TROPICS mission.” *Acta Astronautica* 161: 166–182.

[4] Chia, W.C., Yeong, L.S., Ch'Ng, S.I., and Kam, Y.L. (2015) “The effect of using super-resolution to improve feature extraction and registration of low resolution images in sensor networks.” *Proceedings of the 7th International Conference of Soft Computing and Pattern Recognition, SoCPaR*: 340–345.

[5] CubeSat Design Specification (CDS) Rev. 13, The CubeSat Program, Cal Poly SLO. Retrieved from https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf; accessed on 14 June 2019.

[6] Denby, Bradley and Lucia, Brandon. (2019) “Orbital Edge Computing: Machine Inference in Space.” *IEEE Computer Architecture Letters* **18.1**: 59–62.

[7] Hacker, T.L, and Sedwick, R.J. (1999) “Space-Based GMTI Radar Using Separated Spacecraft Interferometry.” *AIAA Space Technology Conference & Exposition, AIAA 99-4634,* 28-30 September 1999, Albuquerque, New Mexico, 566–579.

[8] Huynh-Thu, Q., and Ghanbari, M. (2008) “Scope of validity of PSNR in image/video quality assessment.” *Electronics Letters,* **44(13)**: 800–801.

[9] Khattab, M.M., Zeki, A.M., Alwan, A.A., Badawy, A.S., and Thota, L.S. (2018) “Multi-Frame Super-Resolution: A Survey.” *IEEE*

International Conference on Computational Intelligence and Computing Research, ICCIC.
[10] Lee, B.B., Martin, P.R., and Valberg, A. (1989) "Sensitivity of macaque retinal ganglion cells to chromatic and luminance flicker." *The Journal of Physiology,* 414: 223–243.
[11] Lee, Kwan Kit, and Ashok, Amit. (2019) "Surpassing Rayleigh limit: Fisher information analysis of partially coherent source(s)." *Optics and Photonics for Information Processing XIII*, 11136.
[12] Long, Xingyu. (2020) "Image Super-Resolution using and Efficient Sub-Pixel CNN." Keras code tutorial. Retrieved from https://keras.io/examples/vision/super_resolution_sub_pixel/; accessed on 10 October 2020.
[13] Lüdenmann, Jürgen, Barnard, Arno, and Malan, Daniël F. (2019) "Sub-pixel image registration on an embedded Nanosatellite Platform." *Acta Astronautica,* 161: 293–303.
[14] Poghosyan, Armen and Golkar, Alessandro. (2017) "CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions." *Progress in Aerospace Sciences,* 88: 59–83.
[15] Qureshi, S.S., Li, X.M., and Ahmad, T. (2012) "Investigating Image Super Resolution Techniques: What to Choose?" *International Conference on Advanced Communication Technology, ICACT*: 642–647.
[16] Sara, Umme, Akter, Morium, and Uddin, Mohammad S. (2019) "Image quality assessment through FSIM, SSIM, MSE and PSNR – a comparative study." *Journal of Computer and Communications,* **7(3)**: 8-18.
[17] Selva, Daniel; Krejci, David. (2012) "A survey and assessment of the capabilities of CubeSats for Earth observation." *Acta Astronautica,* 74: 50-68.
[18] Shi, Wenzhe, Caballero, Jose, Huszár, Ferenc, Totz, Johannes, Aitken, Andrew P., Bishop, Rob, Rueckert, Daniel, and Wang, Zehan. (2016) "Real-Time Single Image and Video Super-Resolution Using and Efficient Sub-Pixel Convolutional Neural Network." *Cornell University Computer Science, Computer Vision [cs.CV] Archive*; arXiv:1609.05158v2 (2016)
[19] Sprigg, Jane, Peng, Tao, and Shih, Yanhua. (2016) "Super-resolution imaging using the spatial-frequency filtered intensity fluctuation correlation." *Scientific Reports,* 6: 38077.
[20] Wang, Z., Bovik, A.C., Sheikh, H.R., and Simoncelli, E.P. (2004) "Image quality assessment: From error visibility to structural similarity." *IEEE Transactions on Image Processing,* **13(4)**: 600–612.
[21] Yao, Tingting, Luo, Yu, Chen, Yantong, Yang, Dongqiao, and Zhao, Lei. (2020) "Single-Image Super-Resolution: A Survey." *Lecture Notes in Electrical Engineering,* 516: 119–125.