JOURNAL OF
EMERGING INVESTIGATORS

Article

Deep residual neural networks for increasing the resolution of CCTV images

Pranav Bantval[1], Parsa Akbari[2]

[1] Canyon Crest Academy, San Diego, California

[2] Downing College, University of Cambridge, Cambridge, UK

SUMMARY

Images from closed-circuit television (CCTV) cameras are often stored with poor resolution due to technical and data storage limitations, reducing the utility of these images. In this study, we propose that informed design and training of an enhanced deep residual network (EDSR) can increase the resolution of CCTV images. This work is the first published application of the EDSR methodology for increasing CCTV image resolution. We utilized a dataset of 986 CCTV images that trained and tested an EDSR for generating high-resolution images from a low-resolution dataset. Implementing an EDSR model, we demonstrate that with 60 layers, performing image super-resolution is extremely effective, producing a peak signal-to-noise ratio value of 76.83 decibels. Furthermore, this technique may be used more generally as a technique for file compression enabling the storage of images at smaller file sizes at low-resolution and the ability to reproduce the same images in high-resolution form.

INTRODUCTION

Video footage has become critical in the resolution of criminal cases and is regularly presented as critical evidence in the court of law (1). However, video surveillance footage is often stored with low resolution or deleted due to data storage limitations. Increasing resolution of video surveillance footage or increasing the length of retention would help resolve criminal cases and investigations. Data compression methodology includes lossless and lossy compression (2). Lossless compression results in reduced file sizes but allows for the original video footage to be regenerated without losing quality or resolution. Lossy compression allows much greater reduction in file sizes; however, the original high resolution video footage cannot be regenerated, therefore lossy compression results in loss of quality (2).

Deep convolutional neural networks (DCNN) are an architecture of models trained with supervised machine learning (**Figure 1**) (3). DCNNs are trained with an iterative process to predict an output from a combination of input features which can include images or videos (3). More recently, DCNNs have been applied as a technique for 'super-resolution' which involves processing low-resolution images or video footage and generating high-resolution output (3, 4). In this process DCNNs are trained with a sample dataset where both low- and high-resolution samples are available for the same footage. The network is trained through an iterative process to generate high resolution footage from low resolution input. The start of the art super-resolution model architecture is enhanced deep residual networks (EDSR), providing significant improvements in model performance by expanding the model size (**Figure 2**) (5). Expanding the model size usually results in reduced stability of training reducing model performance, EDSR networks stabilize model stability by implementing scaling layers that were experimentally validated to be optimal in a variety of input dataset types (5).

Previous research has shown that the application of deep learning for image super resolution has been effective in other contexts such as medical imaging, surveillance, and security (6). However, preceding works have not used the EDSR super-resolution technique in the context of closed-circuit television (CCTV) images. EDSR is the current state-of-the-art model, which could lead to the highest quality single-image super resolution (SISR) produced for security and surveillance purposes (7). We used deep learning with an EDSR model-trained on CCTV images to perform SISR. Previous works of CCTV super-resolution include using a generative adversarial network and nonlinear signal processing, which are both real-time SISR algorithms (8, 9). However, EDSR is the contemporary state-of-the-art DNN, so EDSR should yield higher quality results (10). Image super resolution using a super-resolution convolutional neural network (SRCNN), one of the first deep learning methods to outperform traditional methods by applying CNNs, achieved a peak signal-to-noise ratio (PSNR) of 23.23 dB (11). Typically, CNN is used for image classification, in SRCNN, CNN is used

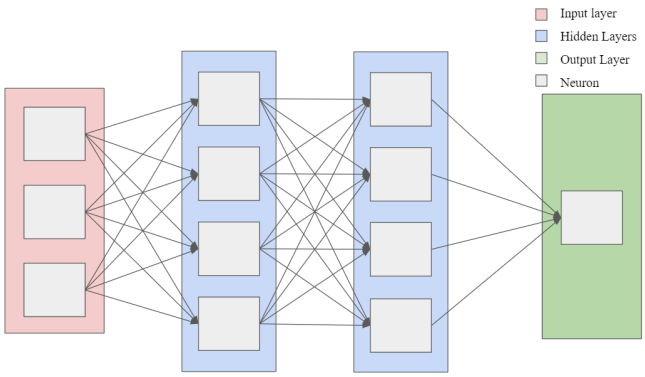

Figure 1: Deep neural network (DNN) example. Neurons are represented by square boxes which process input data and output information to the following layer. The input layer sends the initial data, which is the small image, to the next set of neurons. Each neuron processes the data uniquely and sends its input to the next hidden layer and so on until the final layer where the output is formed. Only the input and output send which is why the layers in between are called hidden layers. The arrows are the flow of the output from the previous neuron to input for another neuron in the next layer.

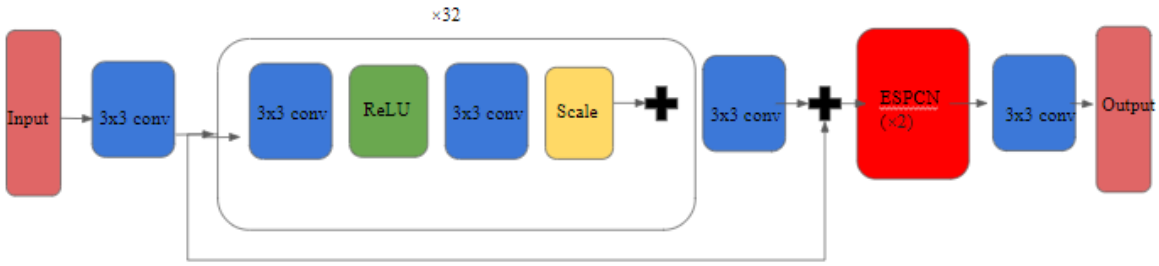

Figure 2: Deep architecture of Enhanced deep residual network (EDSR) Model. The EDSR architecture includes multiple 3 by 3 convolutional layers (3x3 conv) interspaced with rectifying linear units (ReLU) activation functions. Further layers include ESPCN which is an efficient sub-pixel convolutional layer and scale to indicate the pooling process in a CNN.

for SISR. More recently EDSR has been shown to provide superior performance (5). In this study, our model was trained and tested using 986 CCTV images that were shrunk to lower resolution to create a low-resolution training set. We utilized an EDSR model to regenerate the original high-resolution images from the low-resolution training data. Our results show that as the layers of the EDSR model increase, the performance continues to increase past 60 layers.

RESULTS

The performance of our model greatly depended on the values of the hyper-parameters, especially in this case since no one has used EDSR to create high-resolution images from CCTV images and there are no guidelines for selection of model parameters in this context. The purpose of our work was to discover the values of the parameters and model architecture that would produce the best result from the EDSR model. We experimented with the model architecture and training techniques by modifying the initial parameters and model architecture in order to achieve the highest performance possible through an EDSR model. We trained the EDSR algorithm with 15 different parameters (such modification to parameters including changing the learning rate, batch size, and optimizer), primarily increasing the layers by 5 in the range 25 to 60. Across the 8 versions, the performance of the model usually increased as the layers increased (**Figure 3-4**). An addition in layers increased the model performance up to 60 layers. As the layers are increased to 60, it is likely that the model now needs more data and a greater number of iterations to train to good performance as well as more computational power. The Adam optimizer with beta parameters 0.9 and 0.999 used as well as a learning rate between 0.0001 and 0.00025. The batch size remained at 4 images per batch. As the layers increase up to 60, the PSNR value also increases, meaning that the optimum number of layers has yet to be reached.

First, the model is run on a training set, meaning that it takes a set input, which is the low-resolution, 200 x 200 pixel image, and attempts to produce the desired output, which is the high-resolution, 400 x 400 pixel image. After producing the output, it then makes adjustments based on gradient descent. The gradient descent calculates the magnitude and direction of adjustment to the model parameters and then applies a small change in each iteration to minimize the cost function. With each iteration, this action is repeated in order to improve model performance. Model performance is measured by the cost function,

$$J(\theta) = \sum_{i=1}^{m} \vee y_i - h_\theta(x_i) \vee$$

a calculation that includes model accuracy and other metrics of performance. The cost function can be used as a measurement for how many errors the model makes, and

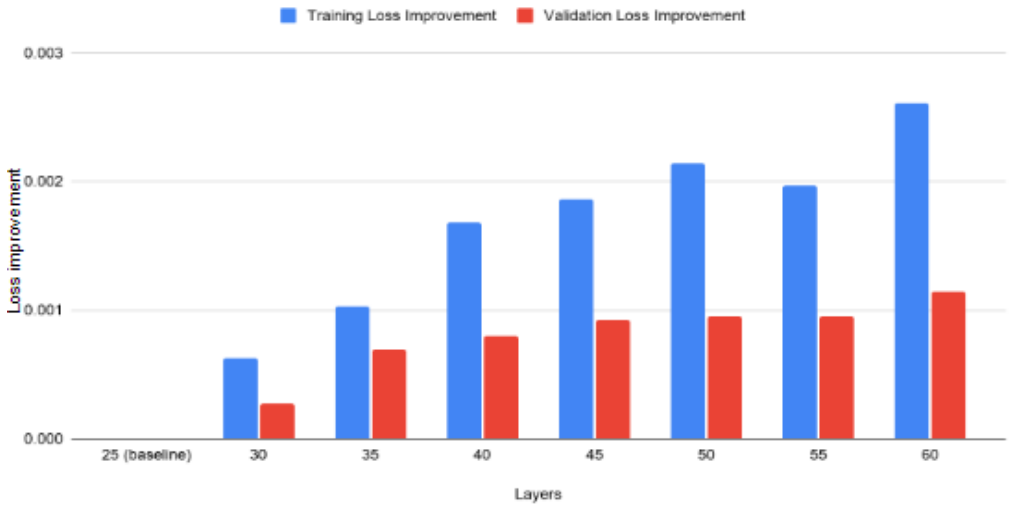

Figure 3: Performance trends over layers. Compared to baseline an increase in the number of layers results in improving validation loss showing the benefit of deeper architectures in model performance. Loss is calculated using the L1 cost function mentioned.

Figure 4: Peak signal-to-noise ratio (PSNR) trends over layers. Similar to improvements in loss as model depth is increased, peak signal-to-noise ratio (PSNR) representing the quality of the super-resolution transformation increases with model depth.

the objective is to minimize the output of the cost function by improving the performance of the model. The output of the cost function is the training and validation loss. The performance is improved by running the model through the training set repeatedly with adjustments based on the gradient descent. The cost function also takes into account the complexity of a model. It is preferred if the model complexity can be reduced with minimal depletion to the accuracy. The cost function prevents overtraining and it allows for a more diverse model.

PSNR is one of the two most commonly used measures to quantify image quality and, in addition to the cost function, was used to evaluate model performance. The term PSNR is an expression for the ratio between the maximum possible value of a signal and the power of distorting noise that affects the quality of its representation. PSNR is used in this experiment to express the results of image super-resolution in quantitative terms. If it is shown that a set of parameters can enhance a degraded, known image more closely to the original, then it can be concluded that it is a better set of parameters. Measuring PSNR and training and validation loss improvement and comparing them between trials provided as an accuracy measurement. Training and validation loss is the error while training or validating the model, measuring the change in loss allows visualization of how much the model is improving between trials.

We found that the most significant influencing factor was the number of neural layers. Changing this affects model performance drastically as a greater number of layers allows the model to infer a greater range of properties from the input image. Improvement can be seen as the difference between training and validation loss and the PSNR values, both increase (**Figure 5-6**). However, as the number of layers increases the model is more likely to overtrain or struggle to optimize the increased number of parameters which must be trained within these additional layers. From these results we conclude that the performance of the EDSR algorithm is highly dependent on the construction of the model. For example, the number of layers in the model is a critical component since the model can analyze smaller details with more layers, producing a higher quality image. However, the downside to more layers is that the model will get extremely complex, requiring more data and longer training times as well as increasing the likelihood of overtraining.

DISCUSSION

In our experiment we demonstrated that an EDSR model can be used to effectively perform SISR on CCTV images. EDSR can be used to generate high-resolution images from low-resolution CCTV images The optimum model performance parameters seem to be increasing the layers

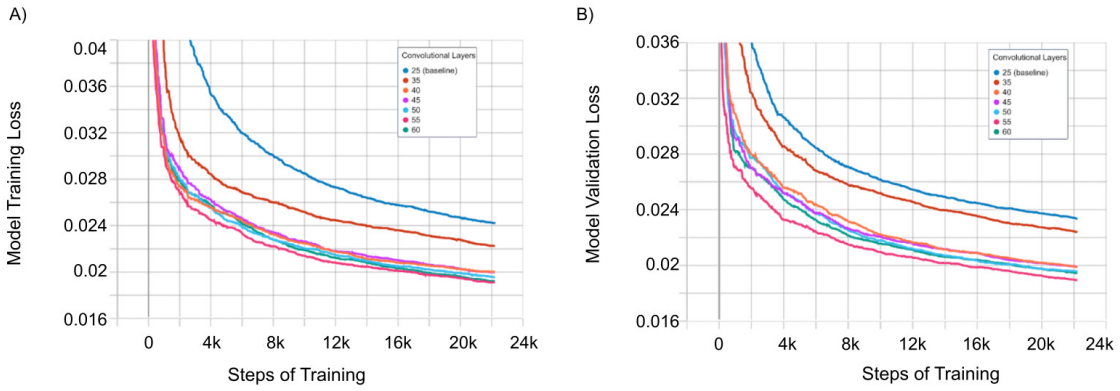

Figure 5: Model training and validation loss over steps. Visualized over steps of training both training (A) and validation (B) L1 loss decrease over time and models with greater depth show higher performance even at earlier time points within the training process. This result shows that despite greater complexity greater depth results in superior models even at earlier stages of training.

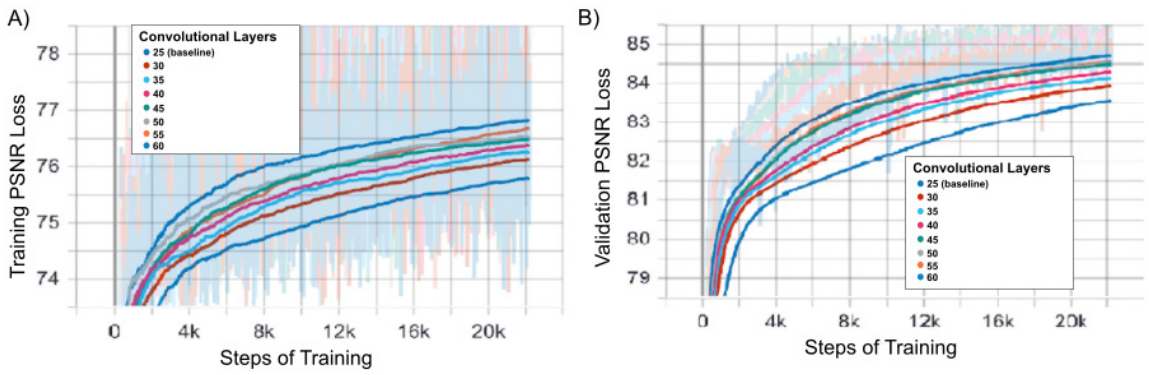

Figure 6: Model training and validation loss over steps. Peak signal-to-noise (PSNR) ratio for training (A) and validation (B) data show a consistent trend of superior performance in models with greater depth at all time points during the training process.

depending on the data set and iterations. With the current resources, it can be concluded that 60 layers is the optimum amount for model performance. There are some issues with pixels that are in very different colors than they should be, possibly due to overflow or clipping. This technique can also be used as file compression since storing images at a smaller resolution and enhancing it when needed is possible. Storing images at 200 x 200 resolution uses about 65 KiloBytes (kB) of memory, while storing images at 400 x 400 resolution uses about 250 kB, roughly 4 times greater. The reduction in storage requirements would be dramatic when applied to a video file. Our work demonstrates the promise of deep neural architectures for achieving high quality performance on the task of image super-resolution. To further expand on this work, a larger data set can be used to train the model, including CCTV images collected at day and night times and in different settings. On top of this, a larger number of iterations need to be trained using a dedicated graphics card which would allow for an increase in layers and possibly an increase in model performance furthermore.

MATERIALS AND METHODS

Previous approaches for CCTV super-resolution have included real-time SISR algorithms (12). Here we utilized a EDSR approach for CCTV image super-resolution, a method which has not been previously utilized for this problem. EDSR algorithms require a significant amount of computational power. An L1 cost function:

$$J(\theta) = \sum_{i=1}^{m} \vee y_i - h_\theta(x_i) \vee$$

was used to assess model performance with an Adam optimizer with beta parameters at 0.9 and 0.999, and learning rate equal 0.0001 to optimize the model, as well as 25 layers (12). Training set batch size was 4 and validation batch size 5, validation was executed every 10 iterations. The analysis resulted in a continual improvement in performance over 20 thousand iterations of training gradually reaching a plateau as iterations of training reached 24 thousand (**Figure 5**) with a training loss of 0.024 and validation loss of 0.023.

First, we used 888 CCTV images of motor accidents for training and an independent dataset of 98 CCTV images of highways for model validation (13). Images from both the training and validation set were scaled firstly to a 400 x 400 resolution, and subsequently to a 200 x 200 resolution. The 200 x 200 low-resolution images were the input for the model, which attempted to replicate the 400 x 400 high-resolution images.

In order to begin training, we used the Pytorch and the EDSR packages in an Anaconda environment (12). Pytorch is an open-source machine learning library used for computer

Figure 7: Super-resolution Image Comparison. A) An original high-resolution image captured from a CCTV camera over an intersection. B) An EDSR high-resolution model generated image showing similar image quality to the original image in Panel A.

vision and natural language processing. The EDSR package is an implementation of the EDSR deep neural architecture. Anaconda is a distribution of Python for scientific programming and is used to add and manage the packages (5, 12). We trained the DNN under a variety of training parameters which include the learning rate, optimizer, batch size of validation and training, epochs, and layers. We utilized the Adam optimizer and varied the learning rate between 0.0001 and 0.00025 (14). The batch size of the training set was varied from four to five, while the validation batch size was consistent at five. The layers of the model started at 25 and increased in intervals of 5 up to 60 where the increase in performance slowed down.

The model was executed over a 6-hour period and the data was saved with SummaryWriter which allowed it to be portrayed as graphs and images in Tensorboard (12). We ran 20 different trials of the model with changes to the stated variables, in order to achieve the best performance. The EDSR model is taking a low-resolution 200 x 200 pixel image and scaling it up to 400x400 accurately by comparing and analyzing the differences between the real high-resolution image and the product of the model (**Figure 7**). Every ten training batches (40-50 images), the model goes through the validation process. The validation process takes another data set with similar images and runs them through the model. After training and validating, we analyzed the data through Tensorboard to conclude the results (12). Succeeding the training and validation is real world application of the trained EDSR model which reads real CCTV images and creates higher-resolution images (**Figure 8-9**).

Received: March 14, 2021
Accepted: February 28, 2022
Published: February 6, 2023

REFERENCES

1. Priambudi, Tegar Kurnia, and Lathifah Hanim. "The Power of Proof against CCTV in Criminal Justice System." *Law Development Journal*, vol. 3, no. 2, July 2021, pp. 193–97.
2. Elakkiya, S., and K. S. Thivya. "Comprehensive Review on Lossy and Lossless Compression Techniques." *Journal of The Institution of Engineers (India)*: Series B, vol. 103, no. 3, Oct. 2021, pp. 1003–12.
3. Alex Krizhevsky Google Inc, *et al.* "ImageNet Classification with Deep Convolutional Neural Networks." *Communications of the ACM*, May 2017, doi: 10.1145/3065386.
4. Choi, Jae-Seok, and Munchurl Kim. "A Deep

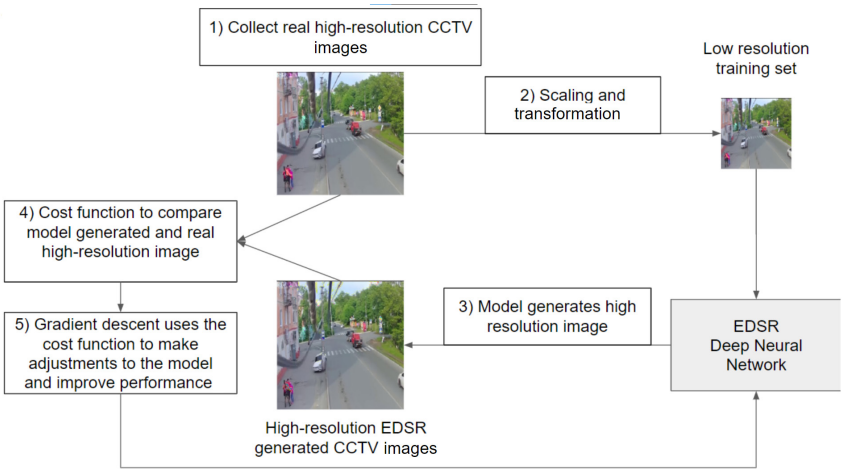

Figure 8: Training an EDSR Model. Our training workflow begins with high-resolution images, which are scaled down to create low-resolution images which are used as training images. The EDSR model increases the resolution of the training data which are then compared to the original high-resolution images using a L1 loss function which then informs gradient descent to optimize model parameters to increase the similarity between model generated and original high-resolution images.

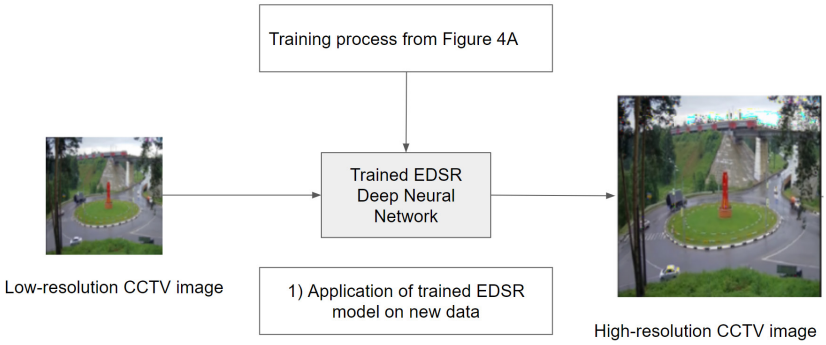

Figure 9: Utilization of a Trained EDSR Model. In comparison to the training process which relies on high-resolution examples for the training process, trained EDSR models can be applied to increase the resolution of images for which high-resolution examples do not exist.

Convolutional Neural Network With Selection Units for Super-Resolution." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 154–60.
5. Lim, B., *et al*. "Enhanced Deep Residual Networks for Single Image Super-Resolution." *Proceedings of the IEEE*, 2017, www.openaccess.thecvf.com/content_cvpr_2017_workshops/w12/html/Lim_Enhanced_Deep_Residual_CVPR_2017_paper.html.
6. Wang, Zhihao, *et al*. "Deep Learning for Image Super-Resolution: A Survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, Mar. 2020, doi: 10.1109/TPAMI.2020.2982166.
7. Yang, Wenming, *et al*. "Deep Learning for Single Image Super-Resolution: A Brief Review." *arXiv* [cs.CV], 9 Aug. 2018, http://arxiv.org/abs/1808.03344. arXiv.
8. Hazra, Debapriya, and Yung-Cheol Byun. "Upsampling Real-Time, Low-Resolution CCTV Videos Using Generative Adversarial Networks." *Electronics*, vol. 9, no. 8, 2020, p. 1312, doi: 10.3390/electronics9081312.
9. Gohshi, Seiichi. "Real-Time Super Resolution Algorithm for Security Cameras." *Proceedings of the 12th International Conference on Signal Processing and Multimedia Applications*, 2015, doi: 10.5220/0005559800920097.
10. Wang, Xintao, *et al*. "Esrgan: Enhanced Super-Resolution Generative Adversarial Networks." *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
11. Zhang, Fu, *et al*. "Image Super-Resolution via a Novel Cascaded Convolutional Neural Network Framework." *Signal Processing: Image Communication*, vol. 63, 2018, pp. 9–18, doi: 10.1016/j.image.2018.01.009.
12. Paszke, A., *et al*. "Pytorch: An Imperative Style, High-Performance Deep Learning Library." *arXiv Preprint arXiv*, 2019, www.arxiv.org/abs/1912.01703.
13. Zaman, Rahat. Highway CCTV Footage Images. www.kaggle.com/insaneshadowzaman/highway-cctv-footage-images. Accessed 31 Jan. 2021.
14. Kingma, Diederik P., and Jimmy Ba. Adam: A Method for Stochastic Optimization. Dec. 2014, doi: 10.48550/arXiv.1412.6980.