CS766 – Mid-Term Report
Super Resolution Image Enhancement

Asher Elmquist (amelmquist@wisc.edu)
Eric Brandt (elbrandt@wisc.edu)

Due: April 1, 2020

1 Overview

The high-level goal of our project is explore image super-resolution and understand how creating higher resolution or higher quality pictures can assist in downstream tasks such as object recognition or image segmentation. The original proposal (see Appendix A) contains detailed information about the relevance of this topic as well as our intended methods of investigation. This report serves the following purposes:

- Report on progress and results achieved thus far
- Explain unexpected challenges and difficulties that have been encountered
- Define remaining tasks and timelines

2 Progress

2.1 Training Data

The training, validation, and test data for our project consists of images from the OpenImages V5 Dataset [1][1]. The full data set consists of an astonishing 9 million images with a combination of human-verified and machine-generated labels. For our training, we decided to focus on training using a subset of these images. After reviewing the labels in the entire dataset, we chose four distinct labels on which to focus: 'Building,' 'Dog,' 'Flower,' and 'Food.' Our hope in narrowing down the scope of image content, we might facilitate future intra- and cross-label inference testing.

After deciding on the appropriate images to use, the acquisition of a data set was broken down into 3 phases: image retrieval, training image preparation, and comparison image preparation.

2.1.1 Image Retrieval

Simply managing the process of acquiring the images was a formidable task. For example, the `.csv` file containing the image identifiers, their corresponding label identifiers, and their web locations is a 3.2 GB file, which itself had to be parsed and searched for images containing the labels of interest. A python script (Retriever.py) was written to perform this task. Because we wanted to focus on large, high resolution images, we used the file size (contained as a field in the `.csv` file) as a heuristic, and only downloaded images that a) had one of our labels, and b) had a file size greater than 5MB. The script was allowed to run for approximately 12 hours, downloading copyright-free and royalty-free images meeting these criteria from the web.

[1]The V6 dataset was released in February 2020. We began our project before that release, and therefore our work is based on the V5 dataset

2.1.2 Training Image Preparation

Once a sufficient number of candidate images having each label were downloaded, we prepared the images using a second Python script (Resizer.py). Using OpenCV [2], this script opened each image and performed the following operations:

1. If the image had a resolution under 2048x2048, the image was discarded from the training set. We are only interested in high resolution images.

2. The image was center-cropped to 2048x2048, to match the input dimensions of our training network. This full-resolution crop was saved.

3. The image was then incrementally downsampled using bilinear interpolation by factors of 2, saving at each resolution. Each image was saved at square resolutions of 2048, 1024, 512, 256, 128, and 64.

We note that at the lower resolutions (e.g. 64x64), the image quality was significantly higher if the scaling was performed in 'steps,' visiting each intermediate power of two enroute to the final resolution. Image quality was much worse if we rescaled directly from the high resolution to the low resolution. Therefore, we settled on the former approach to build our training images.

After this phase, our training data set consisted of the images shown in Table 1.

Label	# of Images
Building	24,530
Food	17,855
Dog	10,736
Flower	24,100

Table 1: Image quantity for each label in our data set

2.1.3 Comparison Image Preparation

The above steps produced ample training data. However, to compare our super-resolution method with 'traditional' upscaling methods, we also generated a comparison data set. To do this, a third Python script (Upsampler.py) was written. This script takes as input a 'starting' and 'ending' resolution. It then traverses the local data set, and upscales all images that are saved at the 'starting' resolution using bilinear interpolation to the 'ending' resolution. The output of this process gives us a point of comparison for our method versus the most widely used traditional method for increasing image resolution.

2.2 Network Architecture and Training Procedure

Based on existing work in image super-resolution, we implemented a convolutional network that utilizes progressive upsampling to produce an image at the target resolution. While other upscaling methods exist, progressive upscaling has been shown to avoid the issues of pre or post-upscaling [3], [4]. Pixel shuffle is used as the upsampling layer as this avoids the some checkerboarding artifacts produced through transposed convolution [5]. The full implemented architecture is shown in fig. 1. The implementation, written using PyTorch [6], includes input and output convolution layers with a ResNet [7] and PixelShuffle block for each doubling of resolution. Optionally, noise can be added at each upsampled resolution to assist in detail creation [8].

The loss used for generating the priliminary results shown in this report uses mean squared error (MSE) loss between the generated image and the target image. While this reduces the the pixel error, it often results in a lack of high frequency detail. To mitigate this issue, we will introduce a second term in the loss function to additionally minimize the content loss follow [9].

For training the network, we use the data sets previously discussed. For the preliminary results, we used 24000 images from our building data set. Since we generated intermediate results, we used those to train the network progressively. That is we trained the network to learn 64x64 to 128x128 super resolution. Once

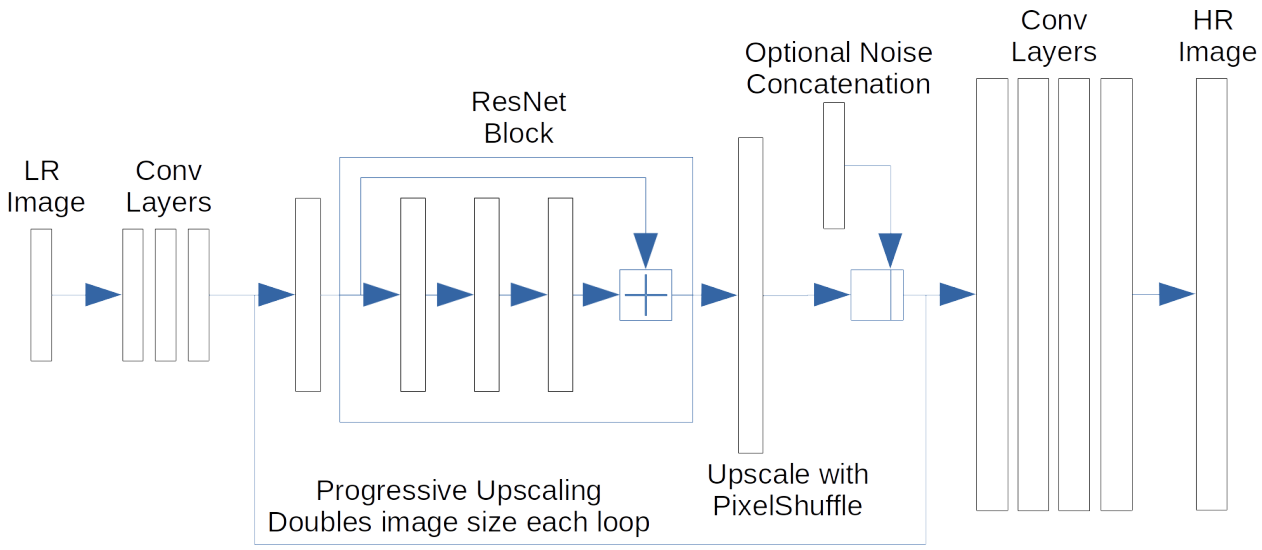

Figure 1: *Flowchart illustrating the neural network architecture used for super-resolution. The network is based on best practices from existing image super resolution and image translation networks.*

the training had converged, we retained the trained weights and introduced another upscaling block. The pretrained layers continue to be trained for higher resolutions, but the progressive growing strategy has been shown to reduce training times and improve convergence [4], [10].

2.3 Website and Code Repository Status

The project code as described above resides in the Github repository located at
`https://github.com/elbrandt/CS766_Project`.
The project website has been updated to reflect current progress and is located at
`https://elbrandt.github.io/CS766_Project/`.

3 Inference Results of Initial Training

Figures 2 and 3 each contain two example images from our food and building data sets respectively. Each began as a high resolution image (2048x2048). It was progressively downsampled to 64x64, shown in the left-most column. The low-res image was then used as input to the trained Super-resolution CNN model to be up-sampled to 512x512. The output appears in the second column. The third column contains the results of a bilinear interpolation upsampling of the low-res image for comparison. Finally, the right column contains the ground-truth image.

4 Challenges / Difficulties

4.1 Data Set Generation

Finding a copyright-free source of high quality, accurately labeled images turned out to be more difficult than we originally thought. Most online data sets available online for purposes of deep learning training typically suffer from either a) being low resolution, b) being high resolution, but not being labeled, or c) being extremely narrow in scope. Image sets of high resolution images that are already accurately labeled are typically copyright, or the terms of use expressly forbid high-volume downloads for the purposes of machine learning.

The data set we found and used does satisfy both our requirements (high resolution, and labeled). However, the data set still suffers from two issues:

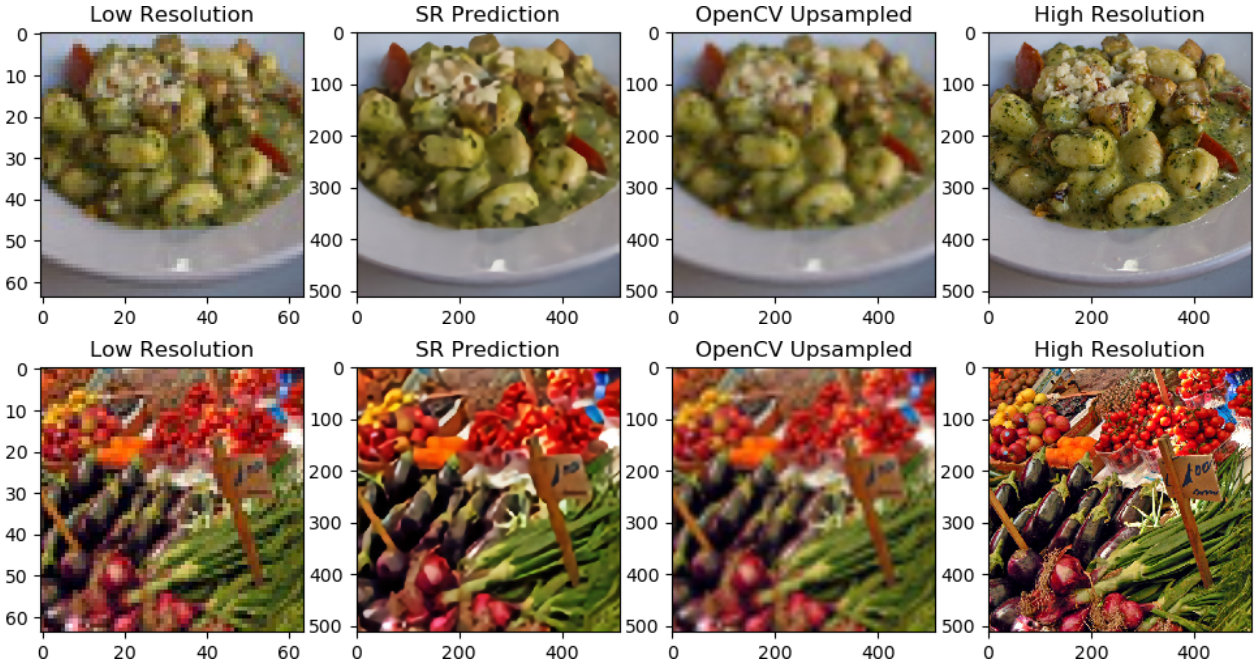

Figure 2: *Two example images from the food dataset (model trained on buildings). Left-to-right: Low-res input, Super-resolution prediction, bilinear upsampling, original high-res image.*

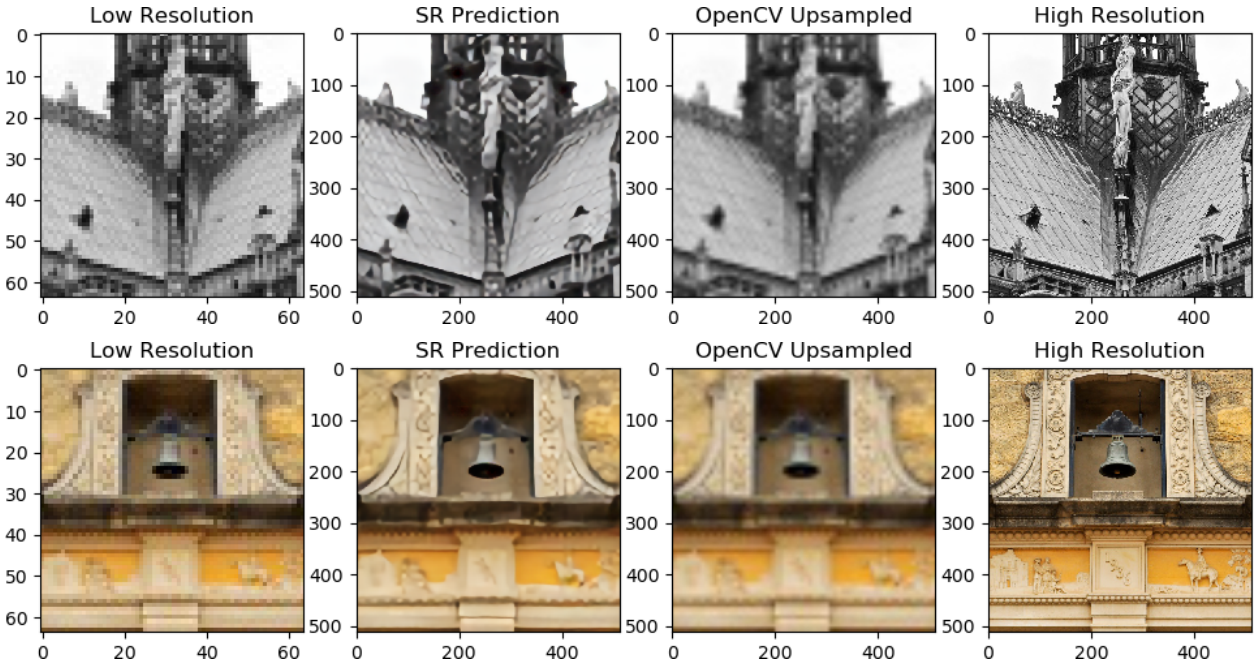

Figure 3: *Two example images from the buildings dataset on which the model was trained. Left-to-right: Low-res input, Super-resolution prediction, bilinear upsampling, original high-res image.*

1. The data set is not uniformly high resolution. Images of all resolutions are mixed in, and the resolution is not known until the image has been downloaded.

2. While the labeling on the images is generally accurate, for many of the images we downloaded, the label is not unique nor the primary subject of the image. For example, images labeled 'food' would ideally

all be images that one might see on a menu at a restaurant: a plate of food and nothing else. However, our data set 'food' images include such things as 'raw materials of food' (e.g. fields of corn), pictures of restaurant façades, scenes of people eating at a party, and of course, some pictures specifically featuring prepared food as the primary and only subject of the photograph.

The first issue can easily be addressed by simply ignoring images below a certain resolution. The second issue is much harder - in that it typically would require manual inspection of each image to determine if an image met our own subjective criteria of what we had in mind when we chose the label 'food.' This is intractable for the quantity of images we are using.

This limitation of 'label-subject quality' in our data set may have significant impact on the ability to evaluate the impact of 'cross-label' versus 'same-label' inferences. For example, we may not be able to effectively evaluate whether a CNN trained on images of 'buildings' does better or worse at inferencing other 'building' images than it does at inferencing 'food' images.

4.2 Trained Model Inference

While our results thus far appear to be closer to the high resolution image than bilinear interpolation, our results do have some small artifacts. Primarily, our results lack the high-frequency detail seen in the HR image. This is in line with previous state-of-the-art SR networks. The proposed addition of content loss should improve this issue.

Another issue which we can easily accommodate in our project, but may prove difficult in a real application, is the resolution and aspect ratio dependence on the input images. We have chosen to train exclusively on square images whose resolution is a power of 2, because this works nicely with our CNN structure. However, in the real world, camera sensors are rarely at this aspect ratio nor are a power-of-two resolution. It is unclear from existing literature how to generalize a trained network to perform on images of arbitrary input resolution.

5 Remaining Tasks

Based on the initial project proposal and the adjustments made as a result of our preliminary results, the remaining tasks are as follows,

- improve the training by augmenting the loss function to include content loss based on VGG
- perform domain specific super-resolution study to quanitify effect of image content on super-resolution
- setup image segmentation for SR benchmark

5.1 Re-alignment of Goals

For the remainder of the project, we have realigned our goals based on further research and understanding after implementation of initial networks. The original goal of understanding domain specific super-resolution will be maintained, but may not be as significant based on the data set challenges discussed earlier in the report. Additionally, we will shift from using object detection for understanding if we can predict sub-pixel object positions to using image segmentation to understand if super-resolution can improve sub-pixel prediction of object boundaries. We will use a pre-trained segmentation network to evaluate and understand these results.

The readjusted goals are now as follows,

1. Implement super-resolution network based on current research in the field
2. Study domain-specific super resolution to understand the effect of image attributes on SR
3. Perform image-segmentation to understand the improvement SR can give for predicting sub-pixel object boundaries

5.2 Performance Metrics

The results shown in this progress report are preliminary and are for visual inspection only with no quantitative performance metrics yet implemented. For the final report and the remainder of the project, we will use two primary metrics. The first metric is a scoring measure used for the 2017 NTIRE super-resolution challenge, found here: `https://data.vision.ee.ethz.ch/cvl/DIV2K/`. The second metric will be based on image segmentation. Segmentation, based on pre-trained models, will be used to evaluate the accuracy of the SR results in predicting sub-pixel object boundaries. We will use the high resolution segmentation results as ground truth. In addition to providing an accuracy metric, this will allow us to understand and evaluate SR as a method for the application of higher resolution segmentation.

5.3 Revised Timeline

Based on the re-alignment of goals, our timeline going forward is as follows:

- **April 1st:** Project Mid-Term Report Due.
- **April 8th:** Finish revision and tweaking of models, begin large-dataset training. Begin evaluating and preparing performance metric test.
- **April 15th:** Begin evaluating performance. Begin working with image segmentation models.
- **April 22nd:** Compile final tests and results data. Begin work on video presentation.
- **April 29th:** Assemble project report, web page, and video presentation.
- **May 4th:** Project web page & video presentation due.

A Original Proposal

The original proposal, as submitted, is included here for reference.

A.1 Project Overview

The purpose of this project is to explore image super-resolution and understand how creating higher resolution or higher quality pictures can assist in downstream tasks such as object recognition or image segmentation. To understand the effect of super-resolution, we propose implementing a convolutional neural network for super-resolution based on the current state of the art. Beyond recreation of a current algorithm, we will study the general nature of the trained model, and explore the application of super-resolution in object detection accuracy and precision. This document covers the relevance of the project, a brief outline of the state of the art, a detailed description of the project plan, and a timeline that we plan to follow in order to accomplish the outlined tasks.

A.2 Project Relevance

Generating super-resolution images from low resolution images has been used in medical imaging [11], [12], astronomy imaging [13] and security imaging [14]. Where small or blurry objects need to be identified, a higher resolution image may increase the performance of existing object recognition algorithms. If super-resolution techniques deliver on their promises efficiently, we can transparently substitute lower resolution or more highly compressed images for costly high resolution images. Image storage, network transmission and video encoding/decoding are several examples of the utility of this. Super-resolution is an interesting task in and of itself, but we propose this project as a way to better understand and explore the potential for super-resolution networks to occupy one step in a pipeline for object detection or semantic segmentation tasks. The ability to locate objects potentially with sub-pixel precision in an image has interesting future applications for photogrammetry and metrology. To this end, our proposal centers on the implementation and application of super-resolution.

A.3 State of the Art

A recent and comprehensive overview of state of the art super-resolution algorithms and network structures can be found in [14]. A short discussion of key points as well as important notes will be given here as they relate to the project proposal. This discussion will cover generative adversarial networks (GANs) versus supervised learning for super resolution as well as a well-studied network structure that has been shown to work well in image enhancement architectures: ResNet [7].

The problem of super-resolution (SR) is in the non-uniqueness of a high-resolution (HR) image generated from a lower-resolution (LR) image. For any LR there are multiple plausible HR that would be faithfully represented by the LR. One way to generate data for SR is to down sample an HR image. Unfortunately, generating training samples from down sampling can lead to small artifacts in the network when trained to directly undo the down sampling algorithm [14]. One way of circumventing this is to use an unsupervised learning approach so as not to unwittingly compute a mapping from input-output samples, but instead to compute a mapping from a distribution of inputs to a distribution of outputs. Generative adversarial networks accomplish exactly this and have been shown to work well for SR [15]. While GANs would improve SR on real-world low-resolution images, we can explore SR in a more efficient manner when we have more control of the datasets and training as is the case with the supervised approach. State of the art results are shown from a supervised method in [16].

In super-resolution, the network architecture plays an important role in the accuracy and efficiency of the network. Many image processing network architectures (including SR) are built on a well-studied convolutional network called ResNet. This network uses a recursive structure to learn small changes in the image. The network is made up of residual blocks with a fraction of the input added directly to the output of a later block. These connections are known as skip connections and are a fundamental component of the ResNet architecture [7] to reduce the complexity of the loss surface and reduce convergence to local minima. These residual blocks, along with up-scaling via deconvolution form the basis of many SR network architectures [14] and are detailed in the implementation given in [16].

A.4 Project Plan

The project will be split into four sections that will allow exploration beyond the state of the art. The first part of the project will be to recreate state of the art super-resolution results using machine learning. In itself this is a difficult task, but there is significant room to explore beyond what has been discussed in the previous section. To go further, we propose studying the generalization of the trained network by training and testing across different image domains. That is, we will train on a set of images, say buildings, and then evaluate the network on an animal image data set. In addition, we propose applying state-of-the-art, off-the-shelf object detection networks such as YOLO-V3 [17] and SSD [18] to understand if object recognition can be improved using super-resolution. Time-permitting, we will then extend to understand if other image improvement could help with object detection such as denoising or sharpening. Since similar networks for super-resolution can be modified for general image enhancement, this is a natural continuation of the project. Finally, we hope to pay attention to the time required to perform SR on images of particular sizes to evaluate the efficacy of real-time SR for live-acquisition applications on either general-purpose or dedicated hardware. Further details and data plans are laid out in the following sections.

A.4.1 Implement and Train Super-Resolution Network

Following the state of the art implementations, we will use a residual CNN as it has been shown to reduce error propagated across layers when given sufficient skip connections. To limit training time, we will avoid using generative adversarial networks. This will result in requiring the network to be trained through supervised learning, meaning we will require input and output image pairs. We do not foresee this being an issue as we can generate a dataset from a single set of high resolution images. This will also give us flexibility in the resolutions we choose to understand the extent to which resolution can be increased without unintended artifacts. For verification, our super-resolution can be compared against ground truth (original image) and bicubic or bilinear interpolation of the downsampled/corrupted image. We can also qualitatively compare our results to state-of-the-art results from the papers discussed previously.

A.4.2 Run Domain Transfer Study

We propose studying the general nature of our network by having three distinct datasets. These may be, for example, buildings, cars/roadways, and animals, each of which have a large corpora of readily available data. We will then hold out one dataset during training to use during testing to see if there are domain specific artifacts that appear on images with significant content differences. We will apply this holdout to each subset in turn to understand the full effects.

A.4.3 Apply Object Detection Networks

To understand if super-resolution improves object detection, YOLO-V3 and SSD will be used with pretrained weights. These networks have been shown to give state-of-the-art object detection accuracy. The accuracy of the pretrained networks will be run on the lower resolution images to generate a baseline accuracy for comparison. The networks will then be run on the higher resolution (cropped when necessary to create one-to-one comparison) to understand if super-resolution improves detection rate, position-precision, or class confidence. Multiple data sets will also allow us to understand if there is significant difference when detecting difference object classes.

A.4.4 Further Extend To General Image Enhancements

If time permits, we can extend this project to look at other image enhancements for improving object detection. Since the residual CNNs have the same architecture as general image processing networks, the network proposed can be adjusted and trained to denoise, or sharpen images at the lower resolution. We would then perform the same studies discussed above on the resulting denoised or sharpened images.

A.5 Web Page

A project web page that will track progress and summarize results has been set up at

- `https://elbrandt.github.io/CS766_Project/`.

All materials related to the project will be tracked in the GitHub repository located at

- `https://github.com/elbrandt/CS766_Project`.

A.6 Timeline

In order to meet the class deadlines and adhere to the project plan, the following timeline is proposed.

- **February 14th:** Project Proposal Due
- **March 13th:** Complete Super-Resolution Implementation
- **March 20th:** Complete Domain Transfer Study
- **March 25th:** Project Mid-Term Report due *Possible re-alignment of goals based on progress thus far*
- **April 3rd:** Complete Object Detection Study
- **April 17th:** Complete Additional Image Enhancement Study
- **April 27-May 1st:** Project Presentations
- **May 4th:** Project Webpage Due

References

[1] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *IJCV*, 2020.

[2] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, "A fully progressive approach to single-image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 864–873.

[4] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[5] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize," *arXiv preprint arXiv:1707.02937*, 2017.

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[9] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, Springer, 2016, pp. 694–711.

[10] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[11] C.-H. Pham, C. Tor-Díez, H. Meunier, N. Bednarek, R. Fablet, N. Passat, F. Rousseau, and H. ene Meunier, "Multiscale brain MRI super-resolution using deep 3D convolutional networks Multiscale brain MRI super-resolution using deep 3D convolutional," *Computerized Medical Imaging and Graphics*, vol. 77, 2019. DOI: `10.1016/j.compmedimag.2019.101647`. [Online]. Available: `https://hal.archives-ouvertes.fr/hal-01635455v2`.

[12] M.-I. Georgescu, R. T. Ionescu, and N. Verga, "Convolutional neural networks with intermediate loss for 3d super-resolution of ct and mri scans," *ArXiv*, vol. abs/2001.01330, 2020.

[13] H. Zhang, P. Wang, C. Zhang, and Z. Jiang, "A comparable study of cnn-based single image super-resolution for space-based imaging sensors," in *Sensors (Basel)*, vol. 19, MDPI, pp. 32–34.

[14] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, vol. 21, no. 12, pp. 3106–3121, 2019.

[15] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[16] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.

[17] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.