Kathmandu University

Department of Computer Science and Engineering

Dhulikhel, Kavre

A Project Proposal

On

"Inpaint Network"

[Code No: COMP 206]

(For the partial fulfillment of 2nd Year/ 1st Semester in Computer Science)

Submitted by:

Utsav Maskey (28)

Manish Bhatta (08)

Binod Sujakhu ( )

Submitted to:

Department of Computer Science and Engineering

Submission Date: 12th November, 2019

Abstract

We propose “Inpaint Network”, a Computer Vision / Deep Learning focused image editing tool that aims at synthesizing photo-realistic images from user-desired manipulation (i.e. re-positioning, removal) of image objects (segments). We intend to provide an interface that facilitates easy re-arrangement of visible objects in a given content photo. The proposed algorithm consists of semantic image segmentation [1], interactive interface for re-arrangement of the segments and image restoration steps [2], [3]. Our segmentation module focuses on masking of input image to its corresponding image-object labels. The user-interface module extracts image-objects from the mask and allows simplified GUI for its manipulation. Our image-reconstruction module analyzes this unprocessed image and attempts to interpolate missing information. Our approach leverages the use of Machine Learning techniques to assist photo editing tasks. The project focuses on using integrated image processing operations for content-aware interaction of visible objects.

Keywords: *Deep Learning, Computer Vision, Image Processing.*

Contents

List of Figure

Figure **Page No.**

List of Tables

Table **Page No.**

Acronyms

CV	**Computer Vision**
CNN	**Convolutional Neural Network**
GAN	**Generative Adversarial Network**
UI	**User Interface**
RAM	**Random Access Memory**
OS	**Operating System**
CPU	**Central Processing Unit**
GPU	**Graphics Processing Unit**
VRAM	**Video RAM**

Chapter 1: Introduction

"Inpaint Network" is a Machine Learning assisted image-editing tool that provides an interface for designers to easily manipulate visible objects in an input image. This tool implements Neural Network models to extract as well as improve contents in the photo.

1.1 Background

Deep Learning methods have dramatically influenced most sectors of Machine Learning including speech recognition, visual object recognition, object detection and many other domains [4]. Ever since the breakthrough by Krizhevsky et al. [5], different forms of CNN have primarily been a go-to algorithm for most Computer Vision related classification tasks. Even deeper architectures have been trained [6]–[9]. CNNs are now flexible enough to perform cutting-edge CV tasks like image Style Transfer [10]–[13], Image Segmentation [1], [14]–[19], Object Detection [20]–[24], Super Resolution [25]–[28], Image Restoration [2], [3], [29]–[31] Frame Interpolation [32], Image Synthesis [33]–[35]. In this work, we focus on combining *image segmentation* and *image reconstruction* operations to assist photo editing tasks.

Figure 1.1 Content Image and its Segmented Mask

In CV, Image Segmentation is the process of partitioning a digital image into multiple segments (image objects) to simplify an image into something meaningful and easier to analyze. According to Krillov et al. [36], the common types of image segmentation are: Instance Segmentation, Semantic Segmentation and Panoptic Segmentation. Traditionally, classical CV algorithms used Thresholding, Region-based, Clustering-based, Watershed-based and Partial differentiation based methods [37]. These methods are significantly out-performed by CNN models [1], [14]–[18]. The applications of image segmentation is significant in medical sector like X-ray, Magnetic Resonance Imaging, etc. [38].

Image restoration is the operation of estimating clean, original image from a corrupt image. Image reconstruction or inpainting is a form of image restoration that can be described as the problem of mapping from a noisy image to a noise-free image. It is an Image-to-image translation problem. The best classical denoising methods approximate this mapping with cleverly engineered algorithms [30]. Some of such algorithms are Anisotropic diffusion, Linear smoothing filters, Wavelet transform, Block-matching algorithms, etc. These algorithms are used for satellite imaging, astronomical purposes, enhancing quality of camera, etc.

Figure 1.2 Image reconstruction. Picture from [31].

1.2 Objective

- To assist and automate creative tasks with Machine Learning tools.
- To research and model a performance-based neural architecture.

1.3 Motivation and Significance

The field of Computer Vision is shifting from statistical methods to deep learning / neural network models. These methods have recently gained popularity in achieving state-of-the-art performance at most of the challenges related to computer vision. But still, the use of such models for practical applications is an active area of research. The motivation behind studying computer vision is to gain an insight into the integrated methodologies of combining machine learning and information processing in vision. The rise of image processing tools and Machine Learning libraries further motivates the automation of computationally processed vision tasks.

This study holds significant importance: Firstly, it attempts to understand complex structures of objects in an image and allows its manipulation. Secondly, it facilitates an interactive environment for designers and artists to gain control of automated machine learning models to suit their tasks. Finally, it makes a significant contribution to the existing literature as it relates to creative assistant systems.

Chapter 2: Related Works / Existing Works

Image Segmentation. Fully Convolutional Networks (FCN) by Long et al. [19] popularized CNN architectures for image segmentation. Since then, many state-of-the-art follow-up models used different forms of CNNs [14]–[18]. Most of these models use DenseNet [8] like architectures, where they use skip-connection[1] techniques to improve de-convolution steps [1]. We decide on using FastAI[2] implementation of Jégou et al.'s [1] model in our segmentation procedure.

Image Manipulation. We attempt to replicate some common image manipulation methods. Various photo editing applications including Adobe Photoshop, GIMP, Paint.Net, etc. provides an interface for handling graphical tasks. We try to implement this by using open-source imaging libraries.

Image Synthesis. Conditional image synthesis refers to the task of generating photorealistic images conditioning on some input data [34]. Such image-to-image translation problems are often approached by some form of GANs [39]. Park et al. [34] demonstrated a terrific tool that focuses on a specific form of image synthesis, which is converting a semantic segmentation mask to a photorealistic image. Isola et al. [35] investigated a general-purpose solution to image-to-image translation problems using conditional adversarial networks. This model can synthesize day-time image to night-time, edges to photo, Monochrome to color images, etc. The key difference is that we focus on synthesizing the region of image that is only affected by our image manipulation module instead of the whole image.

[1] Skip connection: Concatenation of information from encoding layers to its corresponding sized decoding layers

[2] fast.ai © 2019 is a python based deep learning library built on top of PyTorch

Chapter 3: Procedure and Methods

Overview:

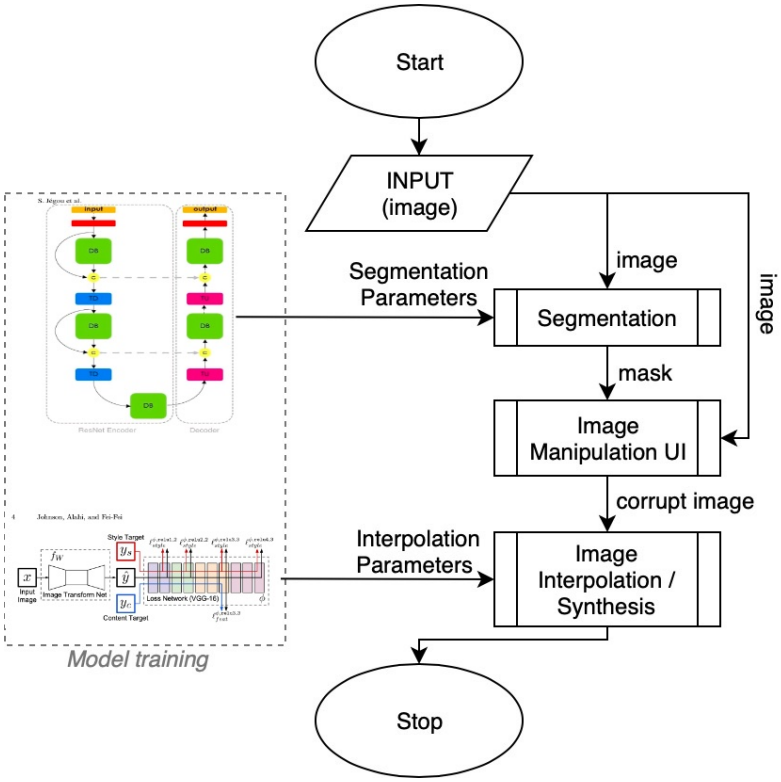

Figure 3.1 Flow chart of the main program. We implement Jégou et al.'s [1] *Segmentation model and Johnson et al.'s [13] image transformation model.*

Our procedure consists of training and inference sub-procedures. Our inference procedure links the trained models to facilitate image manipulation interface.

3.1 Training Phase:

Prior to our testing / inference phase, we train two models that aims at providing Machine Learning solutions for assisted image manipulation.

- Segmentation Model
- Image Restoration / Inpainting

3.1.1 Segmentation Model

We implement Jégou et al.'s [1] model[3] for our segmentation tasks. This approach is based on feed-forward convolutional Encoder path followed by decoder path that maps images to its corresponding pixel-wise class labels (mask). We implement our early encoder or down-sampling path as a transferred version of K. He et al.'s [6] ResNet34 model. Similarly, we link this prior model to a decoder or up-sampling model implemented in FastAI library. Skip connections from the down-sampling to the up-sampling path have been adopted to allow for a finer information recovery for image translation. We use pixel-shuffle [40] as deconvolution layers at upsampling path to match the output labels.

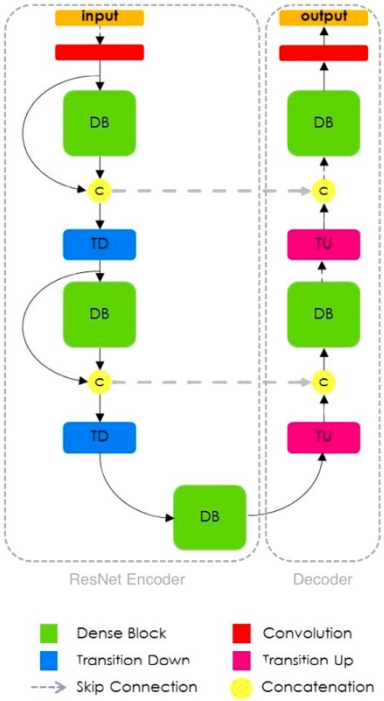

Figure 3.2 Architecture for Semantic Segmentation. Fig from [1].

Architecture details	
No. of layers	34-layer ResNet [6] + Upsampling layers
Activation Function	ReLU [41]
Learning Rate	One-cycle learning rate [42]
Optimizers	• Adam [43] • BatchNorm [44] • Weight Decay
Loss function	Pixel-wise softmax + cross entropy loss
Estimated No. of trainable parameters	~ 20 Million

3.1.2 Image Reconstruction / Inpainting model

We plan on using Howard et al.'s [3] de-crappification model to reconstruct corrupted image portions. This model is based on Johnson et al.'s [13] alternative to GANs that uses a loss network pretrained for image classification model to define perceptual loss functions that measure perceptual differences in content and transformed images. This model is significant for reconstruction as it gives emphasis on low-level image features.

We can further extend our project to attempt various re-construction models [31], [34].

[3] We train both Segmentation as well as inpainting model on CamVid Dataset [45].

3.2 Inference Phase:

At inference time, the program inputs a content image. This image goes through our model to generate corresponding Segmentation mask. The information from this mask is used to facilitate object manipulation. We use OpenCV[4] Mouse events to handle graphical tasks.

Our integrated model reconstructs / inpaints any detected missing information so that re-positioning or deletion of image objects is possible.

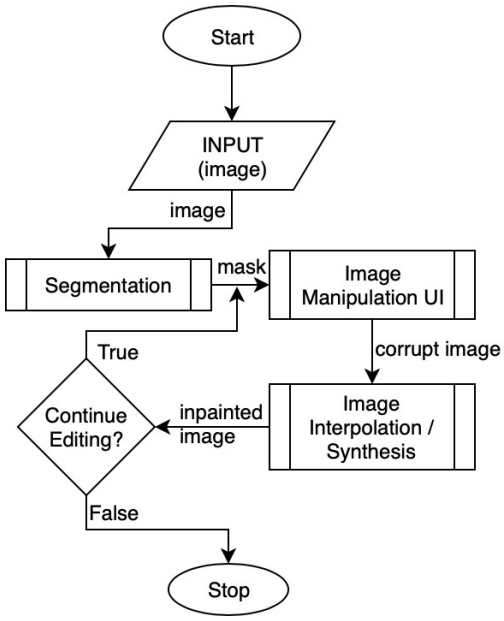

Figure 3.3 Program flow at inference time

This alignment model can be further improved by providing an optional interface for manual Segmentation.

[4] **OpenCV** is an image processing library that mainly aims at real-time computer vision.

Chapter 4: System Requirement Specification

4.1 Software Specification

4.1.1 OS: Linux, Mac, Windows (64-bit)

4.1.1 Python 3.7 Library Dependencies:

Python Library	Version
PyTorch	1.3.0
FastAI	1.0.58
Pillow	5.1.0
OpenCV	4.1.1.26
NumPy	1.17.2

Table 4.1 Python Library Dependencies Table

4.2 Hardware Specification

4.2.1 Training:

- **GPU:** At least 8GB[5] of available VRAM
- **System Memory:** Depends on dataset[6]

4.2.2 Inference:

- **CPU:** 64-bit based processor (Modern CPU recommended)
- **RAM:** About 6 GB of RAM Recommended
- **System Memory:** 2 GB of free space

4.3 Optional Requirements

4.3.1 Web-Cam: Any camera supported by OS (only for external image input)

[5] We train our models on Tesla K80 12 GB GPU provided in Google Colab
[6] 600 MB space for CamVid Dataset [45] + 1 GB space for learned parameters

Chapter 5: Project Planning and Scheduling

This section presents our proposed timings for various tasks of the project. We collectively divide our tasks into five main categories. The work breakdown with time required to complete the specific task are shown as in the Gantt chart below: -

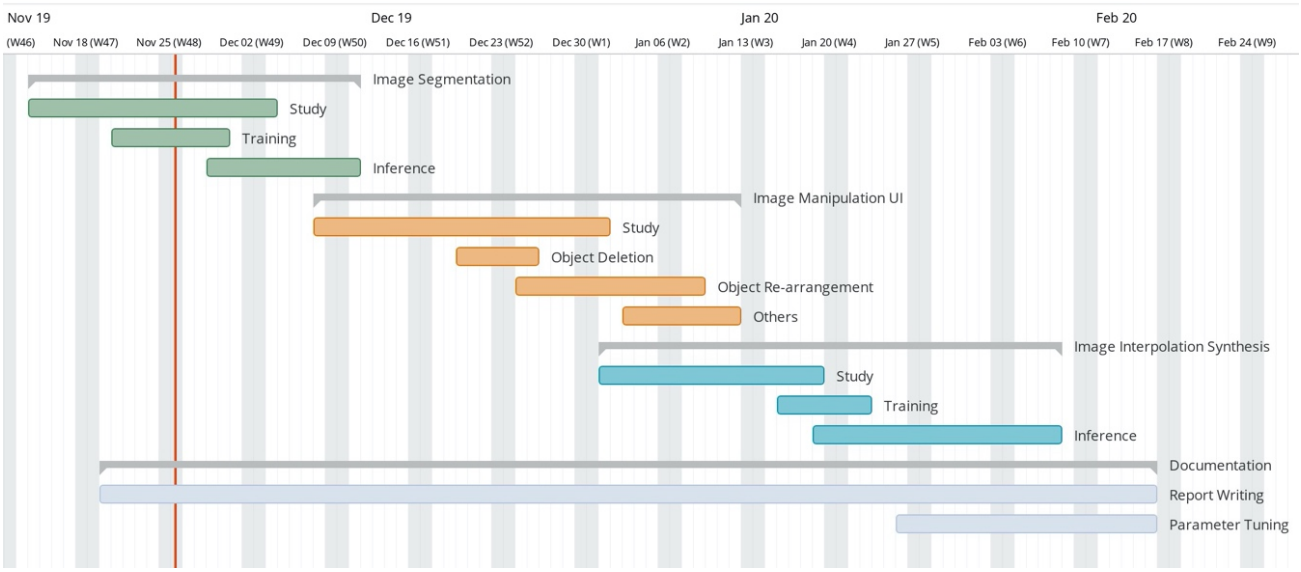

Figure 5.1 GANTT Chart

Legend	
Tasks	**Sub-categories**
Image Segmentation	Study
	Training
	Inference
Image Manipulation (GUI)	Study
	Object Deletion
	Object Re-arrangement
	Others
Image Interpolation / Synthesis	Study
	Training
	Inference
Documentation	Report Writing
Further improvements	Parameter Tuning

Table 5.1 Legend of GANTT Chart

References

[1] S. Jégou, M. Drozdzal, D. Vázquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," *CoRR*, vol. abs/1611.0, 2016.

[2] J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," in *35th International Conference on Machine Learning, ICML 2018*, 2018, vol. 7, pp. 4620–4631.

[3] J. Howard, U. Manor, and J. Antic, "Decrappification, DeOldification, and Super Resolution," in *Facebook F8 conference*, 2019.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, Oct. 2015.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, vol. abs/1512.03385, 2015.

[7] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017.

[10] L. Gatys, A. Ecker, and M. Bethge, "A Neural Algorithm of Artistic Style," *J. Vis.*, 2016.

[11] M.-Y. and L. X. and Y. M.-H. and K. J. Li Yijunand Liu, "A Closed-Form Solution to Photorealistic Image Stylization," in *Computer Vision – ECCV 2018*, 2018, pp. 468–483.

[12] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in *33rd International Conference on Machine Learning, ICML 2016*, 2016.

[13] A. and F.-F. L. Johnson Justinand Alahi, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," in *Computer Vision – ECCV 2016*, 2016, pp. 694–711.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, 2015, pp. 234–241.

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional

Encoder-Decoder Architecture for ImageSegmentation," *CoRR*, vol. abs/1511.00561, 2015.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "PSPNet," *CVPR*, 2017.

[17] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet," *CVPR*, 2017.

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation Liang-Chieh," *arXiv.org*, 2018.

[19] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[21] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *CoRR*, vol. abs/1512.02325, 2015.

[22] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[23] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature Pyramid Networks for Object Detection," *CoRR*, vol. abs/1612.03144, 2016.

[24] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[25] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," *CoRR*, vol. abs/1707.02921, 2017.

[26] W. Shi *et al.*, "Real-Time Single Image and Video Super-Resolution Using an EfficientSub-Pixel Convolutional Neural Network," *CoRR*, vol. abs/1609.05158, 2016.

[27] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[28] W. Yang, X. Zhang, Y. Tian, W. Wang, and J.-H. Xue, "Deep Learning for Single Image Super-Resolution: A Brief Review," *CoRR*, vol. abs/1808.03344, 2018.

[29] S. Lefkimmiatis, "Non-local color image denoising with convolutional neural networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[30] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012.

[31] G. Liu, F. A. Reda, K. J. Shih, T. C. Wang, A. Tao, and B. Catanzaro, "Image

Inpainting for Irregular Holes Using Partial Convolutions," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[32] H. Jiang, D. Sun, V. Jampani, M. H. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[33] Q. Chen and V. Koltun, "Photographic Image Synthesis with Cascaded Refinement Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[34] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic Image Synthesis with Spatially-Adaptive Normalization," *CoRR*, vol. abs/1903.07291, 2019.

[35] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[36] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic Segmentation," *CoRR*, vol. abs/1801.0, 2018.

[37] D. Kaur and Y. Kaur, "Various Image Segmentation Techniques: A Review," *Int. J. Comput. Sci. Mob. Comput.*, 2014.

[38] Z. He, S. Bao, and A. Chung, "3d deep affine-invariant shape learning for brain mr image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[39] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.

[40] A. P. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi, "Checkerboard artifact free sub-pixel convolution: A note on sub-pixelconvolution, resize convolution and convolution resize," *CoRR*, vol. abs/1707.02937, 2017.

[41] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *CoRR*, vol. abs/1803.08375, 2018.

[42] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Residual Networks Using LargeLearning Rates," *CoRR*, vol. abs/1708.07120, 2017.

[43] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." 2014.

[44] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by ReducingInternal Covariate Shift," *CoRR*, vol. abs/1502.03167, 2015.

[45] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence*

and Lecture Notes in Bioinformatics), 2008.