



Traitement d'images à haute résolution grâce à des techniques d'apprentissage en profondeur

Praveer Singh

► To cite this version:

Praveer Singh. Traitement d'images à haute résolution grâce à des techniques d'apprentissage en profondeur. Apprentissage [cs.LG]. Université Paris-Est, 2018. Français. NNT : 2018PESC1172 . tel-02915582

HAL Id: tel-02915582

<https://pastel.hal.science/tel-02915582>

Submitted on 14 Aug 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Thèse de doctorat
de l'Université Paris-Est

Domaine : Traitement du Signal et des Images

Présentée par

Praveer SINGH

pour obtenir le grade de

Docteur de l'Université Paris-Est

**Processing high resolution images through deep learning
techniques**

Soutenue publiquement le devant le jury composé de :

Yulia TARABALKA	Researcher, INRIA	Rapporteur
Sébastien LEFÈVRE	Professor, University of South Brittany / IRISA	Rapporteur
Florence TUPIN	Professor, Telecom ParisTech	Examineur
Renaud MARLET	Director of Research, École des Ponts ParisTech	Examineur
Nikos KOMODAKIS	Associate Professor, École des Ponts ParisTech	Directeur de thèse

Abstract

With the recent advancement in image capturing devices, High-Resolution (HR) images have become quite prevalent under diverse application settings. Fields such as Remote-Sensing (RS) or High Dynamic Range Imaging (HDRI) have been reproducing HR images for past few decades and demand automatic techniques for effectively processing them with least human intervention. Deep Learning has revolutionized these fields owing to parallel processing of a complete batch of HR images in one forward pass. In this thesis we discuss four different application scenarios that can be broadly grouped under the larger umbrella of Analyzing and Processing HR images using deep learning techniques. The first three chapters encompass processing RS images which are captured either from airplanes or satellites from hundreds of kilometers away from the Earth. We start by addressing a challenging problem related to improving the classification of complex aerial scenes through a deep weakly supervised learning paradigm. We showcase as to how by only using the image level labels we can effectively localize the most distinctive regions in complex scenes and thus remove ambiguities leading to enhanced classification performance in highly complex aerial scenes. In the second chapter we deal with refining segmentation labels of building footprints in aerial images. We effectively perform this by first detecting errors in the initial segmentation masks and correcting only those segmentation pixels where we find a high probability of errors. The next two chapters of the thesis are related to the application of Generative Adversarial Networks. In the first one, we build an effective Cloud-GAN model to remove thin films of clouds in Sentinel-2 imagery by adopting a cyclic consistency loss. This utilizes an adversarial loss function to map cloudy-images to non-cloudy images in a fully unsupervised fashion, where the cyclic-loss helps in constraining the network to output a cloud-free image corresponding to the input cloudy image and not any random image in the target domain. Finally, the last chapter addresses a different set of HR images, not coming from RS domain but instead from HDRI application. These are 32-bit images which capture the full extent of luminance present in the scene. Our goal is to quantize them to 8-bit Low Dynamic Range (LDR) images so that they can be projected effectively on our normal display screens while keeping the overall contrast and perception quality similar to that found in HDR images. We adopt a Multi-scale GAN model that focuses on both

coarser as well as finer-level information necessary for HR images. The final tone-mapped outputs have a high subjective quality without any perceived artifacts.

Keywords: High Resolution Images, Deep Learning, Convolutional Neural Networks, Generative Adversarial Networks, Aerial Scene Classification, Building Footprint extraction, Cloud Removal, High Dynamic Range Imaging, Tone Mapping.

Résumé

Grâce aux récents progrès des dispositifs de capture d'images, les images haute résolution (HR) se sont répandues dans plusieurs domaines. En outre, la télédétection (RS) ou l'imagerie à gamme dynamique élevée (HDRI) produisent des images HR depuis quelques décennies et exigent des techniques automatiques pour les traiter efficacement avec le minimum d'intervention humaine. L'apprentissage profond a révolutionné ces domaines grâce au traitement parallèle d'un lot complet d'images HR en une seule passe. Dans cette thèse, nous discutons de quatre scénarios d'application différents qui peuvent être rassemblés dans le cadre plus général de l'analyse et du traitement des images HR par apprentissage profond. Les trois premiers chapitres portent sur le traitement des images RS captées soit par avion, soit par satellite à des centaines de kilomètres de la Terre. Nous commençons par la classification des scènes aériennes complexes, et nous plaçons dans le cadre de l'apprentissage profond faiblement supervisé. Nous montrons comment, en n'utilisant que les étiquettes d'altitude des images, nous pouvons localiser efficacement les régions les plus distinctes. Cela élimine toute ambiguïté et conduit à de meilleures performances de classification. Ceci permet d'éliminer les ambiguïtés et mènent à une meilleure performance de classification. Dans le deuxième chapitre, nous traitons de l'affinement des étiquettes de segmentation des plans des bâtiments, toujours pour les images aériennes. Pour ce faire, nous détectons d'abord les erreurs dans les masques de segmentation initiaux et corrigeons uniquement les pixels de segmentation où nous trouvons une forte probabilité d'erreurs. Les deux prochains chapitres de la thèse portent sur l'application des Réseaux Adversaires Génératifs. Dans le premier, nous construisons un modèle efficace, appelé Cloud-GAN, pour éliminer les couches minces de nuages dans l'imagerie Sentinel-2 en adoptant une fonction de perte cyclique. La fonction de perte cyclique contraint le réseau à produire une image sans nuage correspondant à l'image nuageuse d'entrée, de façon non supervisée. Enfin, le dernier chapitre traite d'un ensemble différent d'images HR, ne provenant pas du domaine RS mais plutôt de l'application HDRI. Ce sont des images 32 bits qui capturent toute l'étendue de la luminance présente dans la scène. Notre objectif est de les quantifier en images LDR (Low Dynamic Range) de 8 bits afin qu'elles puissent être projetées efficacement sur nos écrans d'affichage normaux tout en conservant un contraste global et une qualité de

perception similaires à ceux des images HDR. Nous adoptons un modèle GAN multi-échelle qui met l'accent à la fois sur les informations fines et grossières. Les images générées sont qualitativement remarquables, sans artefacts perçus.

Mots clés: Images à haute résolution, apprentissage profond, réseaux neuronaux convolutifs, réseaux adversaires génératifs, classification de scènes aériennes, extraction d'empreinte de bâtiments, suppression de nuages, imagerie à haute dynamique, cartographie des tons.

Table of Contents

1	Introduction	1
2	Improving recognition of complex aerial scenes using a deep weakly supervised learning paradigm	9
2.1	Introduction	10
2.2	Proposed Framework and Methodology	13
2.2.1	Feature Extraction Module	13
2.2.2	Weakly Supervised Learning Module	14
2.2.3	Object Bank Strategy	15
2.3	Training and Implementation Details	16
2.3.1	Datasets	16
2.3.2	Training	17
2.3.3	Quantitative Results	18
2.4	Results and Discussions	21
2.4.1	Qualitative Results	22
2.5	Conclusions	25
3	Effective building extraction by learning to detect and correct erroneous labels in segmentation mask	27
3.1	Introduction	28
3.2	Related Work	29
3.3	Proposed Framework and Methodology	30
3.3.1	Error Detection	31
3.3.2	Erroneous Label Replacement	32
3.4	Training and Implementation Details	32
3.4.1	Dataset	32
3.4.2	Network Architecture	33
3.4.3	Training	33
3.5	Experimental Results and Discussions	34

3.5.1	Quantitative Results	34
3.5.2	Qualitative Results	34
3.6	Conclusions	37
4	Cloud-GAN: Cloud removal for Sentinel-2 Imagery using a cyclic consistent Generative Adversarial Network	39
4.1	Introduction	40
4.2	Related Work	45
4.2.1	Multi-Model Techniques	45
4.2.2	Uni-model Techniques	46
4.2.3	Generative Adversarial Networks	46
4.3	Physical Cloud Model	49
4.4	Proposed Framework	49
4.5	Training and Implementation	52
4.5.1	Dataset	52
4.5.2	Network Architectures	53
4.5.3	Implementation Details	54
4.6	Results and Evaluation	54
4.6.1	Validation of model design	56
4.6.2	Comparison with state-of-the-art methods	57
4.6.3	Computation-time	59
4.6.4	Results with Dense cloud cover	59
4.7	Conclusions and Discussions	60
5	Deep Tone Mapping for High Dynamic Range Scenes	61
5.1	Introduction	62
5.2	Related Work	66
5.2.1	Tone Mapping Operators for HDR Content	66
5.2.2	CNNs for HDR Scenes	69
5.2.3	Generative Adversarial Networks	70
5.3	Algorithm	72
5.3.1	Problem Formulation	72
5.3.2	DeepTMO (Single-Scale)	74
5.3.3	DeepTMO (Multi-Scale)	76
5.3.4	Tone Mapping Objective Function	79
5.4	Building the HDR Dataset	82
5.4.1	Target Tone Mapped Images	82
5.5	Training and Implementation Details	83

5.6	Results and Evaluation	85
5.6.1	Comparison with the Best Quality Tone-Mapped Images	85
5.6.2	Quality Evaluation	87
5.7	Conclusion, Limitations and Future work	91
5.7.1	Limitations and Future Work	91
5.8	Appendix	93
5.8.1	DeepTMO (Single-Scale) Architecture	93
5.8.2	DeepTMO-R With/Without Skip Connections	93
5.8.3	DeepTMO (Multi-Scale) Architecture	94
5.8.4	Training dataset	96
5.8.5	Dataset Source	97
6	Conclusion and future work	99
6.1	Summary	99
6.2	Future Research Directions	100
	Publications	103
	References	105

List of Figures

2.1	Scenes representing various levels of complexity. (From top to bottom) For low complexity : meadow, bareland, forest; mid complexity: mountain, stadium, parking; high complexity: airport, dense residential, center.	10
2.2	Overall Pipeline	14
2.3	Deep Weakly Supervised Learning (DWSL) network architecture	15
2.4	Scenes from playground and stadium categories	15
2.5	Overall Test Accuracies for different levels of complexities for VGG-16 architecture. Comparisons between VGG-16 (baseline) and VGG-DWSL (our method) clearly show larger scale of improvement in case of mid and high complexity classes	18
2.6	Class-wise accuracies for VGG-16 (baseline) and VGG-DWSL (our method) for the AID dataset. We can clearly see significant difference of improvement in mid and high complexity scenes such as bridges, parking, mountain, stadium, airport and dense residential.	20
2.7	Scenes from Center and Viaduct categories	22
2.8	Qualitative Results from top to bottom for Airport, Bridge, Dense Resi- dential, Mountain, Parking and Stadium. In the first column we have the original image, in the second column we have the most (green boxes) and the least (red boxes) discriminative regions in the original scene, the third column is a visualization by overlaying per-class heatmap over the original image.	24
3.1	Here we showcase predicted segmentation masks for different techniques along with the final ground truth (GT). We can clearly see Resnet50FCN drastically improving over FCN-MLP [105]. Our method (ResnetDR) further improves the fuzzy or "blobby" effects of Resnet50FCN (caused due to naive up-sampling) by simply enforcing the underlying structure of building footprint shapes through learning an error detection and replacement network.	29
3.2	The Network Architecture.	31

3.3	<i>Qualitative Results.</i> The ground truth and predictions are overlaid on top of input image in the last three rows. Red segments are the ground truth, green segments are the predictions and yellow segments represent the overlap of ground truth and predictions. The blue circles highlight the important regions where some significant changes have been found as reported in Section 3.5.2	36
4.1	Cloud-GAN can effectively remove clouds from thin cloudy satellite imagery without supervision using ground truth	41
4.2	Physical Model of a satellite image capture while transmitting through atmospheric clouds	41
4.3	Overall Pipeline of our Cloud-GAN framework	42
4.4	Network Architecture: $Generator_{X \rightarrow Y}$ and $Generator_{Y \rightarrow X}$ represent mapping function $Q : X \rightarrow Y$ & $R : Y \rightarrow X$ respectively. Discriminator X and Discriminator Y represent D_X and D_Y respectively.	44
4.5	Generator and Discriminator architectures including residual blocks.	48
4.6	Synthetic Dataset generation using Perlin noise (a) Reference Image, (b) Cloud Mask (c) Synthetically generated image	51
4.7	<i>Qualitative Results</i> on Real Cloud Dataset. The first row is the input cloudy image while last row is the output from our Cloud-GAN model. All other rows represents results from other state-of-the-art methods.	55
4.8	Comparison of various cloud removal techniques for two synthetically generated scenes. For quantitative evaluation, we show PSNR values which are computed for each image starting from Input and all other methods against the Ground-Truth.	56
4.9	Failure Cases: Over-smoothing or completely fail to produce images for overly clouded images	59
5.1	Comparison between CNN (encoder-decoder) with L_1 -loss and DeepTMO (single-scale). Insets in row 2 show that DeepTMO yields sharp and high resolution output, whereas the CNN results in blurred outputs.	63
5.2	Comparison between CNN (encoder-decoder) with L_p -loss and DeepTMO (single-scale).	65
5.3	DeepTMO Training Pipeline.	66
5.4	DeepTMO (single-scale) generator and discriminator architecture. The generator in (a) is an encoder-decoder architecture. Residual blocks in (c) consist of two sequential convolution layers applied to the input, producing a residual correction. Discriminator in (b) consists of a patchGAN [71, 87, 88] architecture which is applied patch wise on the concatenated the input HDR and tone mapped LDR pairs. More details in Appendix section.	67

5.5	DeepTMO multi-scale generator architecture. While the finer generator G_o has the original image as its input, the input to G_d is a $2\times$ down-sampled version.	69
5.6	Impact of Multi-scale Discriminator and Generator.	71
5.7	DeepTMO (single-scale) with/without FM and VGG Loss.	73
5.8	Batch Normalization vs. Instance Normalization.	74
5.9	Comparison between our DeepTMO outputs and outputs from top-2 ranked tone-mapped scenes on TMQI metrics for a variety of real-world scenes including indoor, scenes with structures, landscape, dark/noisy scenes. In brackets we show corresponding TMQI scores.	78
5.10	Comparison between DeepTMO and targets, highlighting the zoom-ins with the corresponding HDR-linear input.	81
5.11	Quantitative performance comparison of best performing DeepTMO with the target TMOs.	82
5.12	Subjective Test Results. Preference probability of our DeepTMO over best performing target TMOs for 15 scenes representing 5 different scene categories.	85
5.13	<i>Computation time in seconds.</i>	87
5.14	Top TMQI scoring TMOs showing not-so-visually desirable outputs. (a) DeepTMO output, (b), (c) and (d) are 3 top ranking TMO output.	88
5.15	Halo effect. (a) DeepTMO output, (b) DeepTMO trained with log-scaled values, (c) and (d) 2 top ranking TMO outputs.	90
5.16	Color Correction. (a) DeepTMO, (b) and (c) are the color corrected DeepTMO controlled by parameter s from [108].	92
5.17	DeepTMO-R and DeepTMO-S generator architecture.	94
5.18	While the DeepTMO-R simply results in blurred outputs in the bark of tree, the DeepTMO-S tries to refine them but is faced by <i>checkerboard</i> artifacts [50, 120]. The DeepTMO provides best results amongst the three methods while preserving the fine details, contrast and sharpness in the image. . . .	95
5.19	Distribution of best tone mapped output on training dataset of 700 Images.	96

List of Tables

2.1	Accuracies without($m=0$) & with($m=3$) negative instances.	16
2.2	Results comparing baseline with our <i>DWSL</i> method over varied datasets marking state of the art results in all cases. For AID dataset, we have showcased results with both VGG and Resnet architecture, while for all the other datasets we use Resnet architecture due to its higher performance. . .	16
2.3	Performance Comparison Over RSSCN7	17
2.4	Performance Comparison Over AID dataset	17
3.1	Evaluation results on validation set	33
3.2	Evaluation results on test set.	34
4.1	Comparisons for variability in model design in terms of loss function, normalization layer, batch size, and input noise over Synthetic dataset	53
4.2	Quantitative Results showcasing Average PSNR, SSIM and RMSE scores on Synthetic Dataset	53
4.3	Computation Times for different methods	54
5.1	<i>Quantitative Results.</i> Mean TMQI scores on the test-set of 105 images.	84

Chapter 1

Introduction

Deep Learning has gained immense popularity in the recent past owing to its significant improvement in precision accuracies compared to traditional methods for wide variety of computer vision tasks be it large scale image recognition [63, 82], detecting objects of interests [57, 140], pixel level scene segmentation [26, 46] or image processing tasks like image super-resolution [32, 79], de-noising [164, 175], colorization [23, 184] or inpainting [180]. With the availability of high performance Graphics Processing Units (GPUs) and large-scale datasets, it is now possible to train giant models with millions of parameters without any over-fitting, conditioned that the model can easily fit on the GPU memory. This has been a huge boon especially for researchers working with high resolution images in fields such as Remote Sensing (RS) or High Dynamic Range Imaging (HDRI) where applying hand-crafted processing tools on full scale image is both time consuming and scene specific.

RS is the science of observing the earth remotely, either through satellites or from an aircraft, and generating meaningful information from the acquired data. This data can be captured in varied forms, be it Radar, Lidar, MultiSpectral or Hyperspectral depending upon the task at hand. For eg. Synthetic Aperture Radar (SAR) images can be used to produce precise Digital Elevation models of a place [83] or Multi-spectral imagery can be used to detect deforestation [25], examine health of indigenous plants and crops [85], or predict the prospects of minerals [48]. In the last few years, we have seen explosion of petabytes of high resolution dataset freely made available either by government-funded space research programs such as Landsat [158] and Sentinel [43], or through the advent of private players such as Digital Globe and Planetlabs launching their own nano-satellites. In order to deal with such humongous amount of high-resolution satellite/ aerial imagery, there is a dire need for resolving to deep-learning techniques which can easily automatize wide variety of computer vision tasks specific to RS imagery be it land-use classification [18, 98], building [162] / road detection [116], aerial scene classification [66], SAR despeckling [167], hyperspectral classification [107], etc.

HDRI is the technique of capturing full dynamic range of luminance and contrast present in a scene, similar to what is experienced by the human visual system [8, 191]. The human eye, continuously adapts itself, through the aperture of the iris, both to high and low exposure regions in the image. However, this isn't the case in standard capturing devices, where the captured scene is finally quantized within the range of 0-255 pixel luminance value for display purposes. HDRI would be highly desirable for vision applications [137], particularly in cases of autonomous driving or surveillance where both darkest as well as brightest regions hold equal importance while detecting objects of interest. Though there has been several advancements in the past in terms of HDR scene capture [10, 154], HDR display devices are still quite expensive for normal use. With the availability of large-scale high resolution HDR datasets, captured either from high-end HDR cameras or less expensive mobile devices, it is now possible to utilize them for variety of HDR oriented task such as effective quantization (Tone Mapping) [110], Inverse Tone Mapping [39, 41], De-ghosting / Denoising of HDR image [67], etc.

High Resolution can have different interpretations depending on if we are talking about RS or HDRI data. Spatial Resolution in RS imagery largely implies the coverage of ground for each individual pixel in the image. So in a sense a high spatial resolution of 30 cm implies that a particular pixel in the image represents 30×30 cm area on ground. Such a high resolution makes it easier to easily distinguish finer objects such as individual houses or vehicles on ground. A coarser resolution, effectively covering larger area per pixel on ground, makes it difficult to identify clearly such individual objects. The concept of resolution in HDRI is related to the number of pixels per inch. Thus a high resolution HDR image contains larger number of pixels in an image irrespective of the scene area captured.

This thesis, focuses broadly on high resolution images captured either from RS sensors or from HDRI cameras, thus trying to process them by applying various deep learning techniques. In the case of RS imagery, we try to utilize different kinds of deep learning algorithms to solve computer vision tasks such as Aerial Scene Classification, Building Footprint detection as well as removal of thin clouds. For HDRI, we try to address a widely addressed problem of Tone-Mapping but using a deep-learning focused methodology. In the next section, we would briefly discuss about the context and objectives of this thesis, and then finally talk about the larger contributions and the overall structure of the thesis.

Context and Objective

This thesis is broadly focused on three different aspects of computer vision problems in high resolution RS images namely Aerial scene Classification, Building Footprint extraction and Cloud Removal which we try to address by using deep learning methods. Finally we also try

to address image processing tasks in another kind of high resolution images, namely HDR images, by proposing a novel deep learning tool for tone mapping them to a low dynamic range. We hereby illustrate the context and objective of each of the above mentioned task in further details.

Remotely observing the earth can be quite interesting as well as challenging due to the wide spectrum of captured areas of interest (AOI). Compared to natural real world scenes, RS images have much more distinctive objects captured in its AOI. Classification of a scene generally involves, capturing the important regions in a scene, followed by drawing the underlying relationship between these objects which is then used to distill out a meaningful abstract representation of the scene which helps in classifying it. Complex aerial scenes are characterized by large number of such discriminative regions that tend to draw ambiguity in the decision making while classifying them. Annotating these important regions by hand is both time consuming and tedious tasks even for RS experts. To this end, in this thesis, we first try to resolve this confusion with respect to aerial scene classification by proposing a novel deep weakly supervised learning method that automatically localizes the important regions and makes a classification decision using only those regions.

Classifying scene is important as it helps in automatically segregating those class of images from a large dataset which are required for our specific task. For eg. in order to detect buildings in an image, we first need to separate images belonging to building class. Once this has been attained, we move on to segmentations of buildings footprints in such scenes which is highly desirable for wide variety of end goals such as Urban scene planning, monitoring of green cover in an area and emergency relief operations in times of floods, Tsunamis, Earthquakes, etc. With tonnes of satellite imagery available either through google or other sources, it is quite impossible to manually label the building footprints in these images. Recently Open Street Maps have gained huge attraction, however the annotations are error prone and need correction in terms of registration. We try to resolve this task of automatic segmentation of building footprints, by modeling the underlying relationship existing in the joint space of input image and the output segmentation mask. To this end, we propose to detect errors in initial segmentation masks and try to correct them through a replacement technique, both of which are trained end-to-end in a supervised fashion.

Other than recognizing footprints in buildings, RS images are pivotal in wide range of other objectives such as detecting changes in temporally apart scenes, extracting roads in multiple terrains, classification of land cover or land usage. However it quite often happens that these scenes are plagued by films of clouds that partially or completely obstruct the scene. This becomes quite annoying for RS experts especially for cities where the cloud cover is persistent for majority of the year. Thus in order to resolve this, we propose in

this thesis a novel cyclic consistent Generative Adversarial Network model, that generates affectively cloud free images from their cloudy counterparts.

Finally we shift our focus from high resolution RS imagery to processing a fairly different kind of high resolution data, namely the HDR images. We know that our normal display screens are accustomed to project only low dynamic range (LDR) images with 8-bits per pixel (bpp). Thus, in order to display a 32-bpp HDR image on an LDR screen, we require some sort of efficient quantization that can effectively preserve both the overall contrast as well as finer details in bright and dark regions. This technique of mapping HDR content to LDR content while preserving all the desired details is called Tone Mapping. While different sets of Tone Mapping Operators (TMOs) have been designed in the past, their roles have been specific for a particular scene content. We try to utilize freely available HDR images of a wide variety of scene content, to effectively learn a TMO that can yield the most aesthetically pleasing and perceptually good looking tone mapped output through a Conditional Generative adversarial model.

Background on Deep Learning

Deep Learning is a special subset of machine learning which has the capability to learn from observations over a wide variety of dataset (both structured as well as unstructured). These sets of algorithms allow computational models (that are composed of multiple layers of neurons) to learn representations of data with multiple levels of abstraction. While training, these models aims to build a mapping of the intricate structure in large datasets by using the back propagation algorithm [78]. In a way, the back-propagation algorithm allows the flow of gradients backward through the network layers, thus indicating how a machine should change its set of internal parameters to build distinctive representations. These layers are formulated in a sequential fashion such that the representation in each layer is built on top of the representations from the previous layer. Based on the type of learning strategy, deep learning techniques can be broadly classified into 3 major categories: supervised learning, unsupervised learning and reinforcement learning. There are also others which fall at the intersection of supervised and unsupervised learning namely semi-supervised learning, self-supervised learning and weakly supervised learning. In the following subsections, I would introduce each of these in further details.

Supervised Learning Supervised Deep Learning has gained enormous momentum since the seminal paper of Krizhevsky et. al. [82] which showcased a significant boost in image classification task on a large-scale dataset (Imagenet) using deep convolutional neural networks trained on a GPU. Since then, these techniques have been vastly explored in

several domains of computer vision [181], remote sensing [147] as well as medical image analysis [142], to name a few. Supervised learning can be defined as a class of problem where the goal is to learn a mapping between input samples and the target variables, by training the model with some well-annotated examples together with their corresponding ground truths. Several problems such classification, regression, segmentation and object localization can be formulated in a supervised setting.

Unsupervised Learning Unlike supervised learning where input and target is known, unsupervised learning represents a class of problem where a model is learnt that can effectively describe or extract relationships within the data. Several algorithms such as K-means clustering, Restricted Boltzmann machines (RBMs) [146], deep Boltzmann machines [145], Deep belief networks (DBNs) [64] have been explored within the unsupervised framework to automatically learn the representation of the underlying data.

Reinforcement Learning Reinforcement learning represents another class of problem where the goal is to take suitable action so as to maximize reward in a given situation. Reinforcement learning starkly differs from the supervised learning. While in supervised learning, the training data contains the ground truth label for each input sample to train the model, for reinforcement learning, there is no exact ground truth but rather a reinforcement agent which decides the action for a given task at hand. When there is training dataset, the model is bound to learn from its experience. Reinforcement learning has been explored for a variety of problems including control policy making in gaming such as Atari and AlphaGO [117].

Semi-Supervised Learning Semi-supervised algorithms are particular class of algorithms where the model is able to learn from partially labeled data sets. In this setting, the model uses the unlabeled data as input and aim to gain more understanding of the inherent structure in general and build representations. With slow advancement in unsupervised learning, research have made a lot of progress in semi-supervised learning by using tons of dataset from the internet including texts, images, time-series [115].

Self Supervised Learning Self supervised learning is another class of algorithms which drive both from unsupervised and supervised learning. Particularly in this class of techniques, the goal is to convert an unsupervised learning problem into a supervised learning problem by simply creating some pseudo labels from the unlabeled training dataset. In recent literature, these specific class of methods have been getting tremendous attention as they are believed to counter one major limitation of supervised machine learning *i.e.* curating large amount of labelled training data. One recent example of self supervised learning is in

[73] where authors have been able to estimate the relative depth from geometric constraints between the motion of the camera and motion field of the scene, by simply using motion segmentation algorithm. Another example [55], tries to estimate the amount of rotation done as a pre-processing step on the input sample.

Weakly Supervised Learning In deep learning, weakly-supervised learning is performed in those set of problems where the data is noisy, limited, or is collected from an imprecise source. In such conditions, the model is designed to use such weak data to provide supervisory labels for a considerably large amount of training data. Such weakly-labelled learning frameworks have shown great potential in estimating object localization using only image level labels [186].

Contributions

In this thesis, I attempt to solve major computer vision and image processing problems in a broader umbrella of high resolution images that encompass both RS imagery as well as HDR images. The following contributions are presented in this thesis:

1. We present a novel deep weakly supervised learning technique that automatically localizes the most distinctive regions in aerial scenes with image labels. By simply using these important regions, it gains significant improvement in the overall accuracy compared with classification over complete scenes. This contribution has been presented in the following article:

P. Singh, N. Komodakis, Improving recognition of complex aerial scenes using a deep weakly supervised learning paradigm (IEEE Geoscience and Remote Sensing Letters, 2018)

2. We refine the predicted building segmentation masks from a fully convolutional deep learning model, by detecting and replacing pixels with high probability of errors. This is done in an end to end fashion and helps in effectively learning the underlying structure of building footprints thus boosting in the overall Precision and IOU of segmentation labels. This contribution has been presented in the following article:

P. Singh, N. Komodakis, Effective Building Extraction by Learning to Detect and Correct Erroneous Labels in Segmentation Masks, IGARSS 2018 (Oral Presentation)

P. Singh, N. Komodakis, Refining segmentations of buildings and roads through a novel deep structured prediction methodology (to be submitted to Computer Vision and Image understanding Journal)

3. We remove thin clouds in Sentinel-2 imagery by utilizing a cyclic consistent Generative adversarial Network. This we do without any infrared cloud penetrating source (for eg. SAR imagery) or any synthetically generated cloudy-cloud free dataset. We showcase considerable improvement over PSNR values for our own synthetic dataset when performing cloud removal with our technique. This contribution has been presented in the following article:

P. Singh, N. Komodakis, Cloud-GAN: Cloud Removal For Sentinel-2 Imagery using a Cyclic Consistent Generative Adversarial Networks, IGARSS 2018 (Oral Presentation)

P. Singh, N. Komodakis, Removing thin cloud cover in Sentinel-2 imagery using Cyclic Generative Adversarial Networks. (submitted to IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing)

4. We design a novel TMO that can effectively tone map wide range of HDR scenes in order to produce high resolution perceptually good looking LDR outputs. This is done by using a Multi-scale GAN architecture that learns to pay attention to both the global image level details as well as the local finer details. This contribution has been presented in the following article:

P.Singh, A.Rana*, G. Valenzise, F. Dufaux, N. Komodakis, A. Smolic: Deep Tone Mapping for High Dynamic Range Scenes (submitted to IEEE Transactions on Image Processing)*

Structure of the thesis

This thesis is structure into 6 chapters. Each chapter comprises of material from original papers (edited minimally) preceded by a brief summary giving an overview of the following work.

1. Chapter 2 discusses broadly about scene complexity in the perspective of RS images. It gives insights about how complex aerial scenes have large number of discriminative regions some of which might cause confusion to the overall classification of aerial scenes. It lays down technique of how by making the network to learn to select the most important regions in the image can remove this ambiguity and help in improving classification accuracy in complex aerial scenes. It showcases a nice visualizing tool to check for which are the important areas where network lays more weight while classifying a particular scene.
2. Chapter 3 is a step further from classification, as here we talk about semantic scene segmentation for a single class, which in our case is building footprints. The chapter

talks about past techniques and how they are ineffective in generalizing to a wide variety of aerial scenes due to falling short of imposing a well defined structure in their output predictions leading to blobby effects. It henceforth talks about a novel Resnet Detect-Replace model, that while catering to this underlying relationship existing in the joint space of input and output variables, detects the erroneous labels and replaces them with correct segmentation masks.

3. Chapter 4 addresses quite frequent problem in RS imagery, which is of cloud detection and removal. It illustrates first as to how the past techniques either utilize a cloud penetrating Synthetic Aperture Radar imagery (SAR) or paired sets of synthetically generated cloudy-cloud-free images. It elaborates that while SAR is difficult to interpret and of low resolution, synthetically generated cloudy images are far from real. Henceforth it proposes to remove these cloud and in-paint them with the underlying ground structure using a Cyclic Consistent Generative Adversarial Model.
 4. Chapter 5 elaborates a different set of high resolution images, namely HDR images and demonstrates a novel technique for tone mapping them to their LDR counterpart. The chapter shows how past Tone mapping operators have effectively been hand-crafted for specific scene types and require additional parameter tuning. Thus it highlights, how by learning this underlying mapping between HDR and ground-truth LDR images, we are able to yield subjectively superior and aesthetically pleasing high resolution tone mapped outputs. It also showcases how a multi-scale GAN model is able to correct unnecessary artifacts such as blurring or tiling which are quite common in past deep learning based HDR imaging techniques.
 5. Finally the thesis ends with Chapter 6 dedicated to concluding remarks and future works.
-

Chapter 2

Improving recognition of complex aerial scenes using a deep weakly supervised learning paradigm

Aerial Scene Classification (ASC) is the fundamental building block for various complex aerial computer vision tasks such as building detection, semantic scene labeling or vehicle localization. Categorizing highly complex aerial scenes is quite strenuous due to the presence of detailed information with large number of distinctive objects. Recognition happens by first deriving a joint relationship within all these distinguishing objects, distilling finally to some meaningful knowledge that is subsequently employed to label the scene. However, something intriguing is whether all this captured information is actually relevant to classify such a complex scene. What if some objects just create uncertainty with respect to the target label, thereby causing ambiguity in the decision making. We hereby investigate these questions and analyze as to which regions in an aerial scene are the most relevant and which are inhibiting in determining the image label accurately. However, for such Aerial Scene Classification task, employing supervised knowledge of experts to annotate these discriminative regions is quite costly and laborious; especially when the dataset is huge. To this end, we propose a Deep Weakly Supervised Learning (DWSL) technique. Our classification-trained Convolutional Neural Network (CNN) learns to identify discriminative region localizations in an aerial scene *solely* by utilizing image labels. Using the DWSL model, we significantly improve the recognition accuracies of highly complex scenes, thus validating that extra information causes uncertainty in decision making. Moreover, our DWSL methodology can also be leveraged as a novel tool for concrete visualization of the most informative regions relevant to accurately classify an aerial scene. Lastly our proposed framework yields state-of-the-art performance on all the existing ASC datasets.

2.1 Introduction

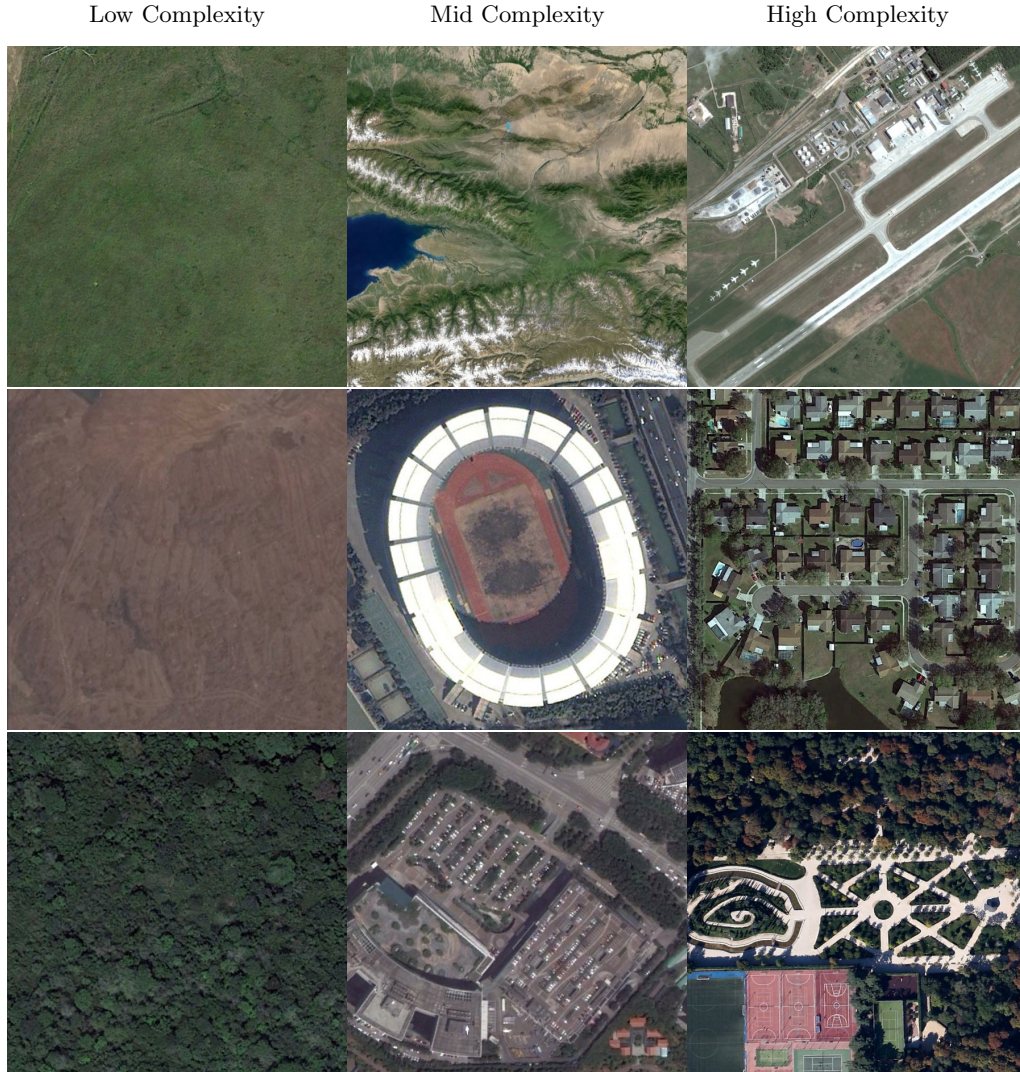


Figure 2.1 – Scenes representing various levels of complexity. (From top to bottom) For low complexity : meadow, bareland, forest; mid complexity: mountain, stadium, parking; high complexity: airport, dense residential, center.

Scene Classification is of paramount importance in remote sensing community to automatically categorize images for further scrutiny. These images are later on utilized by expert annotators for varied roles, including detection or segmentation of objects of interests. For *e.g.* in order to detect building footprints in a dataset of images captured for a city, one primarily needs to recognize scenes that have building structures such as residential or industrial places.

One characteristic feature that holds Remote Sensing (RS) scenes distinctively apart from natural scenes is the widespread scale at which the area of interest is captured. Consequently, the extent of distinctive objects captured in a RS scene are also quite large,

even though, we might not be able to visualize the fine-grained details as it is possible in a natural scene. One of the challenges encompassing RS scene recognition is the added complexity when these discriminative objects are present simultaneously in multiple scenes. One such example is shown in Fig. 2.1, where a scene of Mountain has distinctive patches of arid-land (similar to scene of Bareland) and grasslands (similar to scene of Meadow) however the most representative parts of the scene are the white snowy tracts as would be illustrated in section 2.4. Similarly we have dense trees which are found common in both the scenes of Center and Forest.

Recently deep learning models have shown significant improvement in performance for varied remote sensing tasks such as aerial scene segmentation [6, 165], hyper-spectral classification [90] or change detection [99]. Inspired by architectures of the primate visual cortex [69], these models interpret simultaneously various complex concepts which were lacking in conventional hand crafted methods [178].

The Human Vision is a highly sophisticated system, where a scene is first observed, followed by fetching of relevant information and then formulating an abstract representation. Based upon this understanding, [89] defined scene complexity, based upon how much attention a person devotes to understand a particular scene. For *e.g.* in Fig. 2.1, low complexity scenes such as Bareland or Meadow are easy to interpret while comparatively more complex scenes like Mountain or Center have much more detailed information that needs to be distilled effectively and hence requires fairly larger period of attention.

However, instead of pivoting simply on greater attention for more complex scenes [89], we rather propose an alternate strategy. We argue that by limiting the amount of information gathered from a scene, we tend to minimize the ambiguity at the time of knowledge distillation and thus yield higher performance. Earlier we had seen that the amount of information to be processed differs from scene to scene, with highly complex aerial scenes exhibiting more number of distinctive regions. We postulate that only some of these regions are relevant for characterizing an aerial scene while the other regions are either redundant or irrelevant or inhibit in the recognition performance. This is specifically pertinent in the case of more complex scenes where the number of distinctive regions are significantly large and often lead to confusion in the final decision of a network. We, therefore, propose to remove this ambiguity using a novel methodology which allows the network to learn to choose the most and least relevant regions in a scene that aids in improving the overall recognition accuracy.

Nevertheless, selecting important discriminative regions in aerial scenes is a tedious task. This is mainly because images are of very high resolution and require an expert annotator. We choose to automatize the task of selecting relevant regions by introducing an end-to-end deep weakly supervised learning model in the context of aerial scene classification. Precisely,

we make the network learn to select the most relevant regions in a scene that can predict the scene label with higher accuracy. Since we leverage only the class labels for localizing important regions in our aerial scene without object bounding box annotations, we call it weakly supervised. With further investigation, we also empirically showcase that RS scene recognition using this weakly supervised paradigm is much more beneficial for more complex scenes, thus substantiating our previous hypothesis of ambiguity removal.

In a way, our proposed methodology is a relaxation of the prominent Multiple Instance Learning (MIL) [31] technique where an image is characterized as a bag of instances (region). These bags or images are assigned a label: positive if it contains at least one positive instance and negative if all the instances are negative (Negative Instances in Negative bag or NiN). NiN is a fairly strong assumption simply because for *e.g.* absence of tennis court label doesn't imply that it is absent in the actual scene as seen in Center scene (extreme bottom right) in figure 2.1. It simply implies that the person who labeled the scene based his judgment upon the most predominant region in the scene. Thus, relaxing on the NiN presumption, our network tries to maximize the prediction of the correct class labels by utilizing not only the top K instances (green bounding box in figure 2.8) but even the Bottom M instances supporting the absence of a class (red bounding box in figure 2.8).

Recent work such as [66] has focused upon visualizing the underlying convolutional feature maps of deep learning models by various past techniques like [106]. Through our proposed technique, we also highlight a mechanism to visualize the prediction scores by highlighting the most discriminative regions which help to correctly classify any aerial scene. In a way, it furnishes us with a nice interactive tool to visualize the most important regions in a scene as simple as in a single forward pass through our network. An example is shown in Figure 2.8, where for an Airport scene, the most discriminative regions (runway intersections) are highlighted in green boxes while red boxes indicate negative evidence (fields around) since they confuse it to be a scene of a Farmland.

Lastly, we compare our proposed methodology on all the existing aerial scene classification datasets [173, 178, 192] as well as on a fairly new and challenging one [172], utilizing two prominent deep learning architectures namely Vgg-16 and Resnet-101. Experimental results clearly exhibit our Deep Weakly Supervised Learning (DWSL) method giving state-of-the-art results both with Resnet-101 and Vgg-16 architecture. We also note that Resnet performs notably better than Vgg primarily because of preservation of spatial information throughout the network. This results in much more meaningful information present in discriminative regions of Resnet compared to Vgg.

In a nutshell,

1. We propose the first deep weakly supervised learning technique for aerial scene classification that automatically localizes most prominent regions using scene labels.
-

2. Our model removes ambiguity in complex aerial scenes by selecting only the most important objects out of a large number of distinctive objects that usually confuse the network in overall decision making.
3. We outperform SoA scene classification methods using only few relevant regions thus eradicating the need of utilizing entire scene as done in the past.
4. We present a compact tool for visualizing the most discriminative regions, thus giving us a better understanding of where network focuses upon while classifying a scene.

2.2 Proposed Framework and Methodology

As illustrated in Fig. 2.2, our proposed Deep Weakly Supervised Learning (DWSL) pipeline consists of 3 major steps:

1. Feature Extraction using a VGG16 or Resnet-101 deep learning architecture.
2. Weakly Supervised Learning to select the most discriminative regions and jointly pooling them.
3. Object Bank Strategy to concatenate features from multiple scales and training an SVM classifier on top for ASC.

We discuss each of the steps in more detail in following subsections.

2.2.1 Feature Extraction Module

Feature extraction module is employed to compute a fixed-size feature representation from the input image. We first perform a random scaling of the input image to a size between $[224, 244]$. This is followed by cropping the image to a fixed size of 224×224 . We perform several other data-augmentation strategies like horizontal and vertical flip or rotation by a small angle θ to avoid over-fitting. The resulting image is then passed through the convolution layers of either Resnet-101 or Vgg-16 (whichever is being used for training). Both models have been pre-trained on large scale Imagenet dataset. In Resnet-101 model, the final convolution layer results in an output of size $2048 \times 7 \times 7$ which can be treated as the first block of the Weakly Supervised Learning Module (WSL) (see figure 2.3). In case of Vgg-16, the output from the final convolutional layer is of size $512 \times 14 \times 14$ which is then fed to WSL module as input.

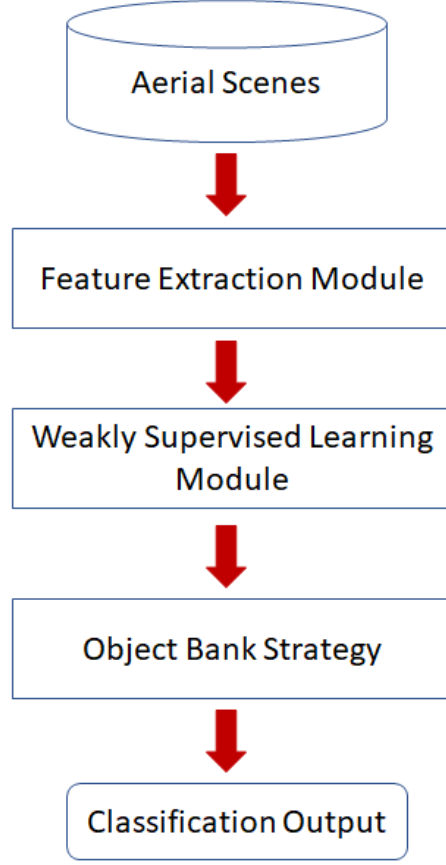


Figure 2.2 – Overall Pipeline

2.2.2 Weakly Supervised Learning Module

Motivated from past works [38, 122, 187] our WSL consists of two major components: Ws1 and K-max M-min Pooling (represented as $s(\cdot)$ in figure 2.3). The Ws1 layer is similar to the fully connected layers and is often witnessed as terminal layers in most deep learning architectures. The only difference is that here we have used them in the form of multiple fully convolutional layers each of size 1×1 . Further, these layers are applied individually to each of the $p \times p$ cells (as shown by the curved arrow) of the previous block. This operation yields us a $p \times p$ block but with a depth size equal to the class number c .

Now, we treat the new output block as stacked heatmaps for each individual class with dimension of $c \times 7 \times 7$ (corresponding to an input image of size 224). These class heatmaps are then fed to a K-max M-min Pooling layer $s(\cdot)$ which (a) selects the top-K scoring regions in a particular class heatmap (highlighted by green cells in figure 2.3, (b) caters to lowest-M scoring regions in the heatmap (highlighted by red cells).

Both top and lowest scoring activations are projected onto the original image and visualized as green and red bounding boxes in Figure 2.8. Finally, these cells are aggregated

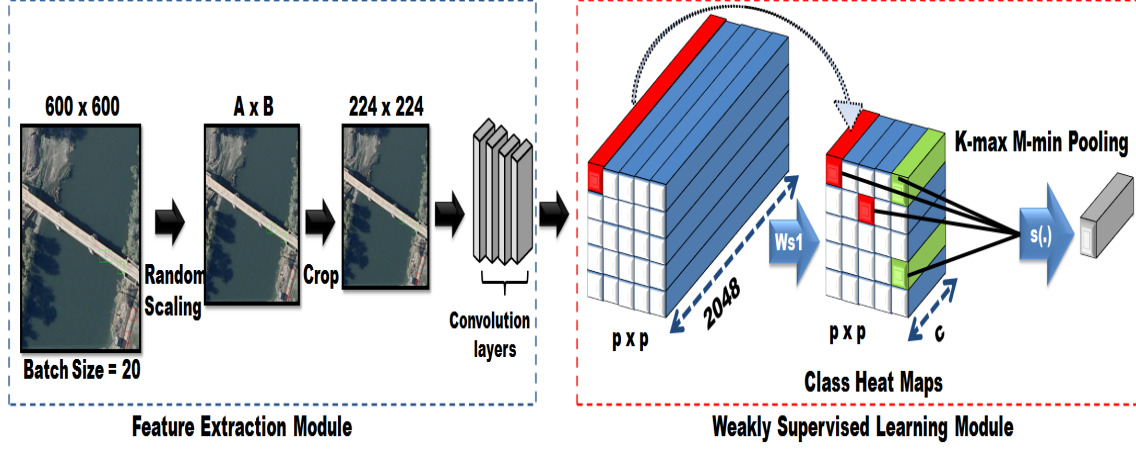


Figure 2.3 – Deep Weakly Supervised Learning (DWSL) network architecture



Figure 2.4 – Scenes from playground and stadium categories

individually for each of the c class heatmaps using $s(c)$ which is given as: $s(c) = s_{top}(c) + s_{bottom}(c)$. The components $s_{top}(c)$ and $s_{bottom}(c)$ are formulated as: $s_{top}(c) = \sum_{k=1}^K t_k(c)$ and $s_{bottom}(c) = \sum_{m=1}^M l_m(c)$. The $t_k(c)$ represents the k^{th} top-scoring activation value for c^{th} class heatmap. Similarly, $l_m(c)$ is the m^{th} lowest-scoring activation value for the same c^{th} class heatmap. We set $K = 3$ and $M = 3$ as default settings for our DWSL model based on empirical results. At the end after applying $s(\cdot)$ operation, we obtain one $c \times 1 \times 1$ layer representing the classification scores for each class.

2.2.3 Object Bank Strategy

As a final step, we adopt an object bank strategy as highlighted in Fig. 2.2. Similar to [38], we concatenate features computed from multiple scales of the input followed by training a support vector machine (SVM) classifier on top. This trained SVM model is then used to classify scenes from test dataset thus yielding final aerial scene classification score.

The top-K and bottom-M scoring regions are equivalent to the positive and negative evidences of a Multiple Instance Learning paradigm [31]. At the time of training, the K-max M-min pooling trains the network weights to accurately localize both the most discriminative regions (positive instances) that can correctly classify the scene plus those

Instances	% OA Resnet-DWSL (without object bank)
k=3,m=0	92.56 \pm 0.22
k=3,m=3	93.02 \pm 0.57

Table 2.1 – Accuracies without(m=0) & with(m=3) negative instances.

regions which have no correlation with the class (negative evidences). We call it weakly supervised learning since the localization is learned only using the global scene variables rather than actual supervised bounding box annotations. Both the dominant regions and the regions supporting the absence of a class can easily be visualized in the overlaid class heatmaps in figure 2.8.

Dataset	Arch.	Baseline	DWSL	% Gains
AID [172]	VGG	91.33 \pm 0.46	95.06 \pm 0.35	<u>4.11</u>
AID [172]	Resnet-101	95.44 \pm 0.26	96.96 \pm 0.34	1.52
WHURS19 [173]	Resnet-101	96.94 \pm 0.92	98.67 \pm 0.61	1.73
RSSCN7 [192]	Resnet-101	94.48 \pm 0.33	96.19 \pm 0.50	1.71
UCMerced [178]	Resnet-101	97.24 \pm 0.35	97.81 \pm 0.26	0.57

Table 2.2 – Results comparing baseline with our *DWSL* method over varied datasets marking state of the art results in all cases. For AID dataset, we have showcased results with both VGG and Resnet architecture, while for all the other datasets we use Resnet architecture due to its higher performance.

The reason for choosing negative instances is that some classes which have small inter-class variability (Playground and stadium class in Fig. 2.4) might result in high classification scores for both by only using positive instances, while classifying a scene lets say of stadium class. This is due to the fact that there is presence of positive instances (playground field and stadium roof tops respectively) for both the classes. In such a scenario, negative instances provide with complementary information (stadium roof giving clear evidence of absence of playground class) thus resulting in correct classification of the scene as stadium. This is elucidated empirically for with and without negative instances for our Resnet-DWSL model without object bank strategy (table 2.1). Boost in overall classification score clearly highlights the necessity of using negative instances.

2.3 Training and Implementation Details

2.3.1 Datasets

We conduct our experiments on 4 different datasets namely, AID[172], UCMerced[178], WHURS19 [173], RSSCN7 [192]. AID is a fairly recent large scale dataset composed of

10000 images coming from 30 different classes, drawn from Google imagery captured using multiple imaging source. An additional level of complexity arises from the larger intra-class variations as each sample is collected from different locations over varied time period and seasons. The other well known datasets UCMerced[178], WHURS19 [173] and RSSCN7 [192] are limited in number and somewhat saturated in terms of performance.

Method	Overall Accuracy (%)
Pretrained GoogLeNet [172]	85.84 \pm 0.92
Hierarchical Coding [171]	86.4 \pm 0.7
Pretrained VGG-VD-16 [172]	87.18 \pm 0.94
Pretrained CaffeNet [172]	88.25 \pm 0.62
Deep filter banks [170]	90.4 \pm 0.6
Two-stage deep feature fusion [94]	92.37 \pm 0.72
Our DWSL method	96.19 \pm 0.50

Table 2.3 – Performance Comparison Over RSSCN7

2.3.2 Training

Our experimental setup is somewhat similar to [172]. We randomly draw 50% of our dataset as training set and rest is kept for testing. We repeat this step *thrice* to avoid the influence of randomness and compute precision accuracy for each run. We report overall mean and standard deviation over all runs in table 2.2. We perform training on two different architectures namely VGG-16 and Resnet-101 as mentioned in section 2.2.

Method	Overall Accuracy (%)
Pretrained GoogLeNet [172]	86.39 \pm 0.55
Pretrained CaffeNet [172]	89.53 \pm 0.31
Pretrained VGG-VD-16 [172]	89.64 \pm 0.36
<i>salM³LBP – CLM</i> [14]	89.76 \pm 0.45
Combining 2 FC Layers [19]	91.87 \pm 0.36
Two-stage deep feature fusion [94]	94.65 \pm 0.33
Our DWSL method	95.06 \pm 0.35

Table 2.4 – Performance Comparison Over AID dataset

For the **baseline models** (VGG-16 and Resnet-101), we first clip the final softmax layer + fully connected layer. Then, we add a fully connected layer with outputs equal to

the number of classes present in the target dataset, finally appended with a softmax layer. Additionally, for baselines, we first pre-train the aforementioned models on ImageNet and then, fine-tune them on target dataset.

For proposed **DWSL models**, we clip the last max pooling layer (just after convolution layers) + all the fully connected layers including soft max layer. Moreover, we simply add the Ws1 layer followed by K-max M-min pooling for Resnet-101 model. In case of VGG-16, an additional convolution layer of $7 \times 7 \times 2048$ convolutions is added before the Ws1 layer. We add a softmax layer at the end for both the models. Our training loss is a log loss function which depends on the probabilities computed by softmax layer.

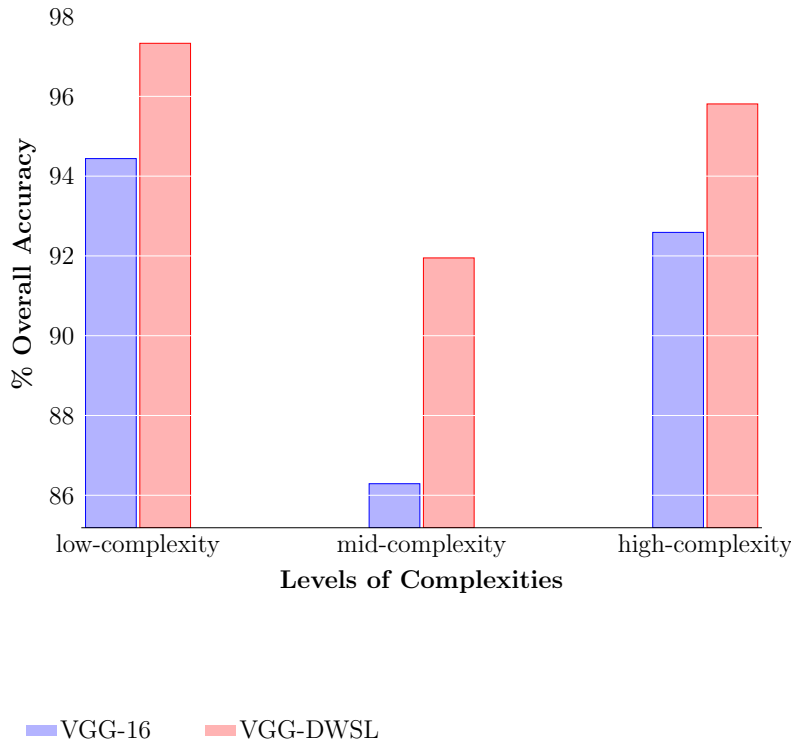


Figure 2.5 – Overall Test Accuracies for different levels of complexities for VGG-16 architecture. Comparisons between VGG-16 (baseline) and VGG-DWSL (our method) clearly show larger scale of improvement in case of mid and high complexity classes

2.3.3 Quantitative Results

We compare the performance of our proposed model (DWSL) with the baseline model through overall mean accuracy over all runs as reported in Table 2.2. Our weakly-supervised learning technique outperforms the current baselines by a considerable margin over all datasets. Especially in the case of VGG architecture, we witness a significant performance gain over the much recent AID [172] dataset.

To effectively fine tune our networks, we utilize a dampening factor layer (with $dt=10$)

to divide the back-propagating gradient just before it reaches the feature extraction module. This is mainly to assure that the pre-trained feature extraction weights are not tampered much. We train our models using Adam optimization technique [80] which computes adaptive learning rates for each parameter over the course of iteration. We set the initial learning rate to $1e - 4$. Each run consists of 50 epochs and we choose the best epoch based upon the test accuracy.

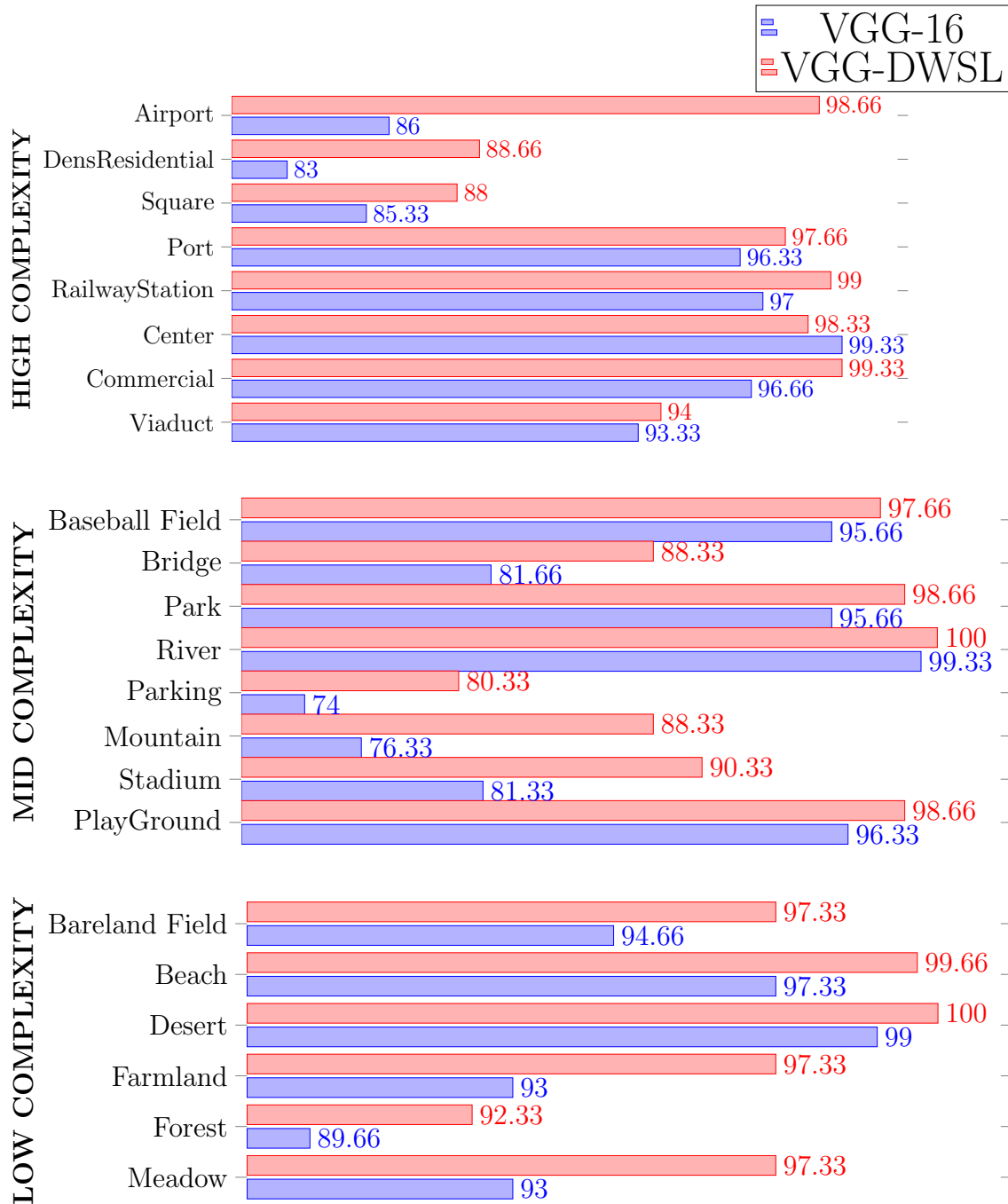


Figure 2.6 – Class-wise accuracies for VGG-16 (baseline) and VGG-DWSL (our method) for the AID dataset. We can clearly see significant difference of improvement in mid and high complexity scenes such as bridges, parking, mountain, stadium, airport and dense residential.

2.4 Results and Discussions

Additionally for completeness, we also show comparisons with other state-of-the-art methods on AID and RSSCN7 benchmark datasets where we outperform all other hand-crafted as well as deep-learning methods by considerable margin. Our results are summarized in tables 2.3 and 2.4. Our method is superior to second best results by a margin of 3.8% and 0.4% on RSSCN7 and AID datasets respectively

Hence, the results clearly emphasize that learning to pay attention to few discriminative regions in a scene and thus learning a correlation between these regions is more beneficial than recognizing by visualizing the entire scene. Another conclusion that can be drawn here is since the design of Resnet architecture naturally preserves spatial information throughout the network, it tends to learn more meaningful discriminative regions as compared to VGG network thus contributing to better performance.

Since Resnet architecture gives state of art performance on [172] which is a fairly challenging dataset, therefore, we stick to Resnet to demonstrate the effectiveness of our method for all other well known datasets.

Next, we study the recognition accuracies from the perspective of scene complexities as depicted in Figure 2.5. We observe that enhancement in performance is significantly higher in case of medium (5.66 point) and high complexity (3.22 point) regions than in case of low-complexity (2.88 point) regions. Thus we can fairly conclude that our weakly supervised methodology broadly assists in minimizing ambiguity while recognizing more complex scenes where there is the presence of much more distinctive regions or what can be simply perceived as richer information.

Delving deeper into class-wise predictions (in figure 2.6), we witness huge improvements in recognizing mid-level or high level complexity scenes such as Bridges (6.66 points), Parking (6.33 point), Mountain (12 point), Stadium (9 point), Airport (12.66 point) and Dense Residential (5.66 point).

Complex aerial scenes are characterized by large number of distinctive objects or detailed image regions and many a times it happens that two or more classes might occur simultaneously in one scene (as illustrated in section 2.1). Thus it becomes quite challenging for a neural network to distinctively identify the correct class to which the scene belongs. Our model removes this ambiguity by making the network to learn to choose the most and the least relevant regions in the scene that are distinctive enough in order to correctly identify a particular class.

An important point to note here is for the accuracy test results for high complexity scenes, we find that most of them have, on an average gain of at least 1.3 - 2 percentage points which is quite significant considering that for many of them the baseline performance



Figure 2.7 – Scenes from Center and Viaduct categories

is already quite high. Only in two of the cases, *i.e.* for center and viaduct the performance becomes comparable. In case of Center class (Fig. 2.7), we observe that most of the scenes in this category have one big roof structure in the center of the image instead of large number of scattered objects throughout the scene. Thus it is more straightforward to classify a scene from the complete image rather than selecting few relevant regions. Similarly in case of viaduct, there are large-scale loopy structures located in the center of image which again makes both our method and baseline that uses full image competing enough.

2.4.1 Qualitative Results

Figure 2.8 demonstrate qualitative results for our methodology by depicting aerial scenes (column 1), most and least distinctive regions marked by green and red boxes respectively (column 2) and the overlaid resulting heatmaps (column 3). It is quite evident that the network selects only those relevant regions which are characteristic feature of a particular class. At the same time it also picks up insignificant regions which best demonstrate the absence of a class.

For eg. in case of the bridge scene, a neural network might confuse it with a river if overlooking at the entire scene. However by providing a weak localization supervision using image labels in our method, we tend to make the network decide on which are the most representative regions in a scene that best describe a scene. Similarly in case of mountain scene, the network, by looking at the entire scene, might interpret it to be a Bareland or Forests or Meadows, however by focusing on just the snow covered regions it tends to correctly classify it as Mountains. Thus we can intuitively interpret that making the network to localize distinctive regions and only focusing upon them for recognizing the

scene, considerably removes the ambiguity caused while visually inspecting the complete scene.

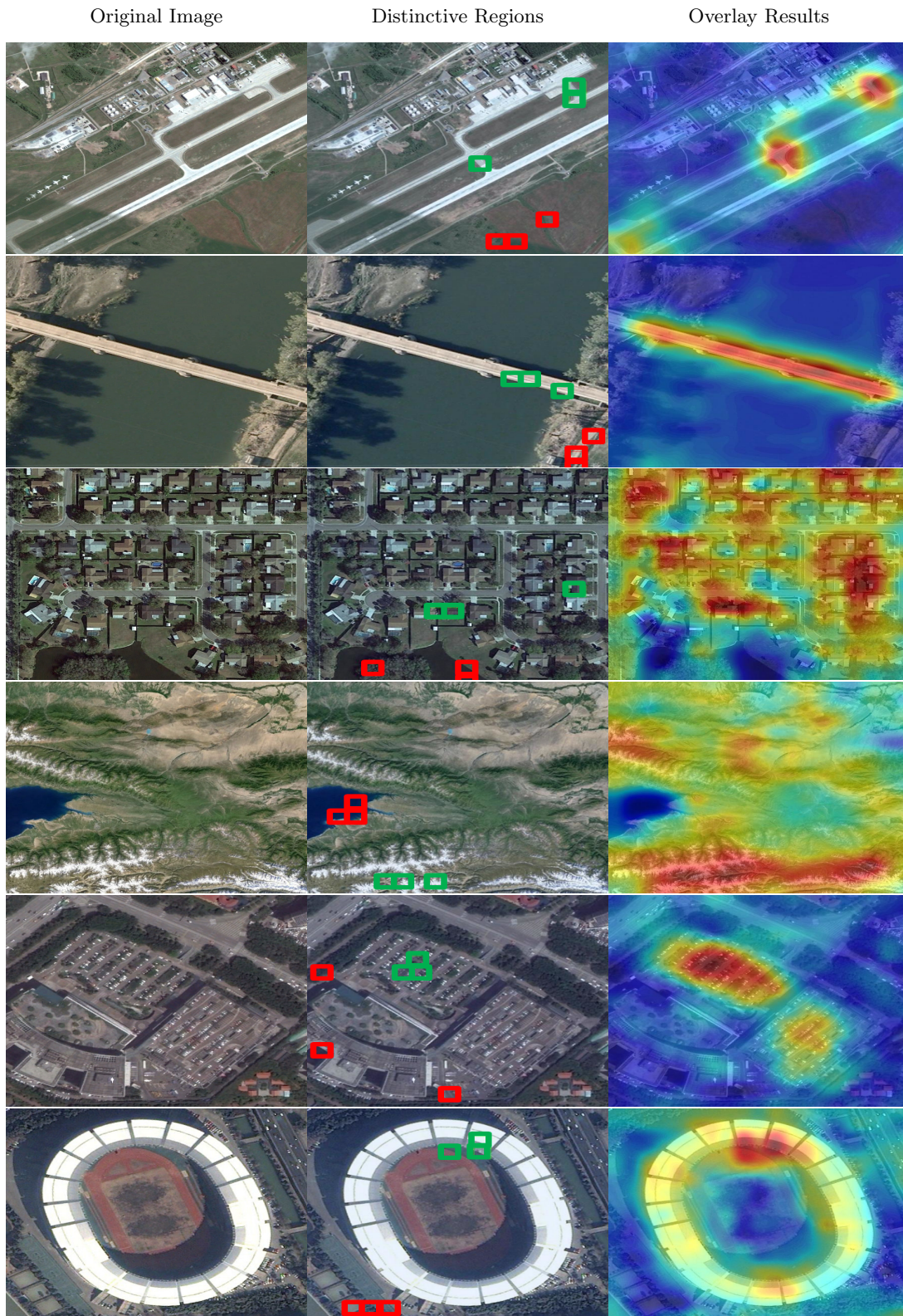


Figure 2.8 – Qualitative Results from top to bottom for Airport, Bridge, Dense Residential, Mountain, Parking and Stadium. In the first column we have the original image, in the second column we have the most (green boxes) and the least (red boxes) discriminative regions in the original scene, the third column is a visualization by overlaying per-class heatmap over the original image.

2.5 Conclusions

In this chapter, a deep weakly supervised technique is proposed in the perspective of aerial scene classification. We illustrate how a network can learn to localize the most predominant regions in an image simply from the scene labels. We also conclude that using these discriminative regions instead of the entire scenes results in state-of-the-art performance while at the same time providing meaningful interpretable information that allows one to elucidate better the decision of a network. We substantiate empirically as to how, by adopting our weakly supervised paradigm, we tend to remove ambiguity in more complex aerial scenes, resulting in a boost in their performance. In addition to this, our experimental results also underline that Resnet architectures suits much better to our task due to its characteristic feature of spatial information preservation. Finally we showcase a nice visualization tool to highlight most relevant regions in aerial scenes.

Chapter 3

Effective building extraction by learning to detect and correct erroneous labels in segmentation mask

In the last chapter we saw how choosing the most distinctive regions in a scene helps in removing ambiguity and improves in complex aerial scene classification. While classifying such scenes is important to determine which aerial images are relevant for a particular task at hand (for *e.g.* to extract building footprints, we need to first classify urban images), the ultimate task is to label the objects of interest in the scene. Thus, semantic scene segmentation is one of the most pivotal features for remote sensing image analysis. Although existing segmentation techniques perform well on similar landscape images, their generalization capability on an entirely different landscape is extremely poor. One of the primary reasons is that they partially or wholly, neglect the underlying relationship that exist in the joint space of input and output variables. Thus, effectively they lack to impose structure in their output predictions which is necessary for successful segmentation. In this chapter, we address this problem and propose a novel solution by modeling the joint distribution of input-output variable which in turn enforces some structure in the initial segmentation mask. To this end, we first detect erroneous labels, in the form of *Error maps*, in the initial building masks. These Error maps are then used to correct the corresponding erroneous labels through a replacement technique. We evaluate our methodology on the benchmark *Inria Aerial Image Labeling* dataset, which is a large scale high resolution dataset for building footprint segmentation. In contrast to previous methods, our predicted segmentation masks are much closer to ground truth, owing to the fact that they are able

to effectively correct both the large errors as well as the *blobby* effects. We lastly perform at par with other state-of-the-arts, validating the efficacy of our technique.

3.1 Introduction

Recently, we have witnessed an explosion of petabytes of high-resolution remote sensing datasets such as Sentinel 1-5 [43], SpaceNet [27] and Inria Aerial Image Labeling dataset [105]. Till now, these datasets have been manually annotated by experts. However, with the availability of such an enormous amount of high-resolution data, it is truly herculean to label all of them by hand. This necessitates the automatic segmentation of these remote sensing images to quickly and effectively detect varied points of interest such as roads, buildings, forests in an image for tasks such as urban scene planning, green cover monitory and other emergency relief operations such as floods, forest fires and cyclones. Deep learning methods have recently shown significant improvements on automatic semantic labeling tasks for classical datasets such as Vaihingen or Potsdam. [4, 5] fused semantic maps from multiple sources through a residual correction technique. Whereas, [103] corrected shifts in OSM maps through an iterative refinement technique using Recurrent Neural networks. In addition, [112] used boundary detections to improve the semantic segmentation and report impressive performance on Vaihingen dataset.

Interestingly, [105] showcased the drawback of models trained over such classical datasets, by highlighting their lack of generalization capability to other cities that are captured under different conditions. They, henceforth build a new dataset covering a much larger surface of the earth including both densely populated urban landscape as well as sparse alpine regions under varied illumination conditions. Subsequently, they split their train and test data such that they come from different cities. Finally they also addressed problem of *blobby-predictions* *i.e.* curvy edges of building masks. By fusing feature maps from different levels of convolution network, both [105] and [104] combine low level edge information with high level semantic rich information. For the same task, [15] uses a multi-task loss in order to approximate the distance transform and the semantic maps.

Lately, most of the aforementioned techniques have partially or fully neglected the idea of enforcing structure in the output label space. This mainly results in these blobby effects due to naive up-sampling during the segmentation masks prediction. On the contrary, a few of those that do enforce structure, either do it through a novice human assumptions about the structure of the output label space (in the form of hand engineered CRF pairwise potentials as in [103]). Or they rely on semantic maps from other sources to refine the initial predicted labels [4, 5]. However, these refinements through residual correction can only correct small errors (such as on the boundaries) while leaving major segmentation

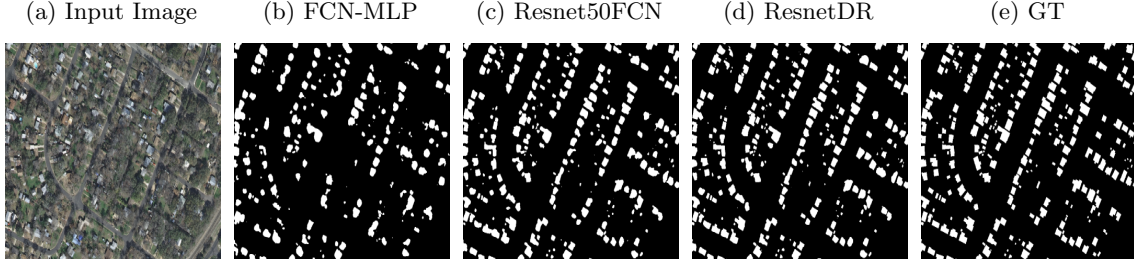


Figure 3.1 – Here we showcase predicted segmentation masks for different techniques along with the final ground truth (GT). We can clearly see Resnet50FCN drastically improving over FCN-MLP [105]. Our method (ResnetDR) further improves the fuzzy or “bloby” effects of Resnet50FCN (caused due to naive up-sampling) by simply enforcing the underlying structure of building footprint shapes through learning an error detection and replacement network.

errors intact.

Built on these observations, we argue that we need to learn a joint structure of input and output variables that can effectively predict the replacements for major segmentation errors and thus rectify them. Adapted from [54], we henceforth, propose a joint input-output model which segments the building footprints in high resolution imagery. Our proposed method is build up of two steps. Firstly, our model predicts an error map EM based upon the input image X and initial segmentation S . We then update the labels in regions of high error probabilities with a new label prediction which in turn relies on X , S and EM for its decision. Through the error map EM , we learn a joint distribution between input and output variables which further helps in enforcing structure in the final label prediction. As shown in Figure 3.1, our model refines the bloby effect by learning the underlying structure of the building footprints which is enforced using the error maps. Finally, this leads our method to perform at par with existing state-of-the-art on Inria Image Labeling dataset [105].

3.2 Related Work

Various techniques have evolved in the recent past for segmentation as well as refinement of building footprints in satellite images. While some employed information from lower-order statistic, others have used Conditional Random Fields (CRF) Potentials or Recurrent Neural Networks to refine the finally generated Segmentation maps. There have also been works utilizing Priors for building shapes or using other open sources for refinement.

First Order Statistics Works falling under this category generally use first order statistics information from an image to train a deep network for the task of semantic segmentation. Bishke et al. [15] showed how by utilizing signed distance as an auxiliary

loss in addition to prediction of segmentation maps, they are effectively able to incorporate geometric information into the internal representation learned by a network. In a way, they were able to preserve the semantic boundary information while predicting the segmentation masks. Maggiori et al. [105] compounded information from all layers of the networks (starting from the first till the last layer), thus utilizing both low level information as well as high level abstract representation to predict the semantic maps.

CRF Potentials or RNN's Works under this category use higher order potentials which generally add smoothing constraints in the form of neighborhood clique's to refine the segmentation maps. Vakalopoulou et al. [162] optimized an MRF model built on top of deep learning features computed over very high resolution imagery. Similarly Maggiori et al. [103] trained a recurrent neural network in the form of an iterative refinement process to correct the segmentation maps over several passes.

Prior Regarding Polygon Shapes Works falling under this category have tried to solve building detection as a problem of predicting the polygonal representation of earth objects. PolyCNN [56] directly learns to predict the coordinates of the geometric shape of an object (vectorial representation) instead of first predicting per pixel labels followed by vectorization. [157] on the other hand try to assign binary labels to the mesh triangles by utilizing the prior knowledge of building edges meeting at right angles as a regularization.

Using other sources for refinement This last category includes works which utilize open street maps (OSM) as a noisy segmentation label for learning to predict aerial scene segmentation. While [76] only relies on OSM for training its deep learning model, [5] fuses representation learnt from OSM and from optical imagery to better predict the aerial segmentation maps. Even though with noise, OSM's provide a useful additional source of guiding tool for the network to make much refined predictions.

3.3 Proposed Framework and Methodology

Lets assume our initial input image to be $X = x_{i=1}^{C \times H \times W}$, where C , H and W are the channel, height and width of X respectively. Similarly, $S = s_{i=1}^{C \times H \times W}$, be the initial segmentation map. Our technique aims to model a joint relationship between input X and output variables (S) to rectify and produce a much more refined version of S . This can be formulated as $S' = G(X, S)$ where S' denote the updated segmentation mask after replacing erroneous labels with new labels.

As shown in Figure 3.2, our proposed methodology comprises of two major steps. First, we predicts errors (EM) occurring in the initial building segmentation mask S with the

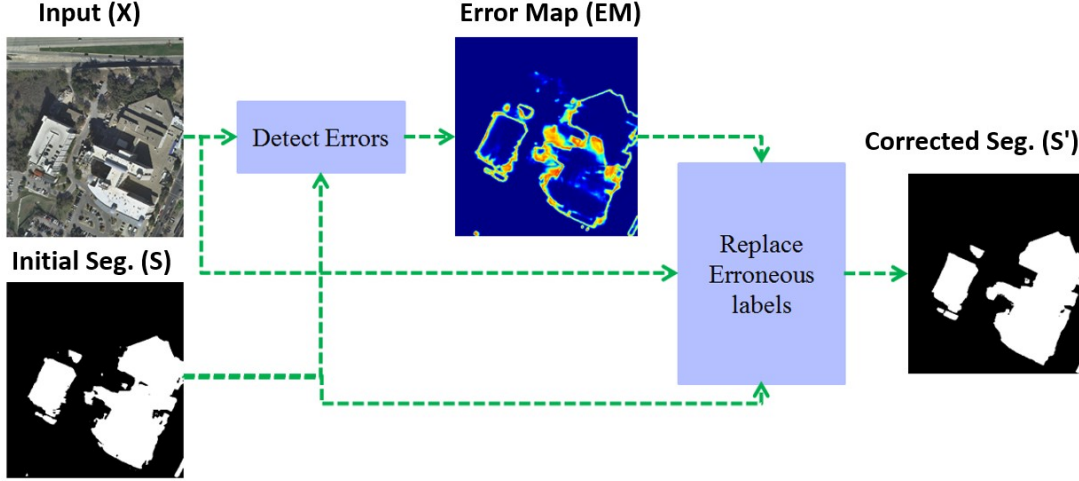


Figure 3.2 – The Network Architecture.

help of input image X . Next, we utilize these error maps to correct those erroneous labels in S , whose error probability is large enough (in EM) by replacing them with predicted labels. In the following, we explain these two steps in detail.

3.3.1 Error Detection

The error detection component (G_{ed}) computes the probability map (EM) to detect the erroneous labels in the initial segmentation mask S , stated as:

$$EM = G_{ed}(X, S). \quad (3.1)$$

In other words, G_{ed} learns from the joint input output space of X & S to predict an error probability score between 0 and 1. In other words, it predicts whether or not a particular label s_i is erroneous, if so, s_i gets replaced with a correct label in the next step. G_{ed} can be easily formulated as a deep neural network which requires as such no explicit auxiliary loss and can be learnt under a single umbrella of one loss between corrected segmentation and ground truth segmentation.

3.3.2 Erroneous Label Replacement

The updated label S' is a convex combination of the initial segmentation mask S and updates from the replacement component denoted by G_{elr} . It is given as:

$$S' = EM \odot G_{elr}(X, S, EM) + (1 - EM) \odot S. \quad (3.2)$$

The error map EM generated from G_{ed} acts as a gateway to restrict G_{elr} so as to just focus on the erroneous labels of S and replace them with the predicted ones. Similar to G_{ed} , G_{elr} can also be modeled using any deep learning architecture. If we restrict the EM probability maps to 0 and 1, the forward pass of Replacement happens as:

$$S' = \begin{cases} S, & \text{if } G_{ed}(X, S) = 0 \\ G_{elr}(X, S, EM), & \text{if } G_{ed}(X, S) = 1. \end{cases} \quad (3.3)$$

This shows that only erroneous labels are being replaced while the non-erroneous labels remain intact. For the back-propagation of gradients we have:

$$\frac{dL}{dG_{elr}(\cdot)} = \begin{cases} 0, & \text{if } G_{ed}(X, S) = 0 \\ \frac{dL}{dS'}, & \text{if } G_{ed}(X, S) = 1. \end{cases} \quad (3.4)$$

In a way, the gradients update G_{elr} only for those regions where erroneous labels are found in S , thus restricting G_{elr} to pay attention and predict replacements only for these particular regions. Additionally, passing Error maps to the G_{elr} component, helps it to rely on the correct labels to predict replacements for the new erroneous labels. Altogether, G_{elr} improves and makes correction of these erroneous labels, by jointly reflecting upon the Error maps, the input X and the initial segmentation S .

3.4 Training and Implementation Details

3.4.1 Dataset

We evaluated our method on the Inria Aerial Image Labeling Dataset [105] which consists of Satellite imagery of urban settlement over the United States and Austria. The entire dataset consists of two classes namely, building and not building. All the images are of size 5000×5000 and have a resolution of 30 cm with RGB bands. First 5 images from each class were chosen for validation, while the rest were used for training. For testing, we submit our test predictions to the project webpage online <https://project.inria.fr/aerialimagelabeling/>.

3.4.2 Network Architecture

We initially train a Fully Convolutional Network [149] (FCN) adapted to Resnet-50 [63] architecture to generate our initial segmentation mask S . This model, which we name as Resnet50FCN, is trained to reconstruct ground truth segmentation masks with a given input image X and ground truth labels. We henceforth treat Resnet50FCN as our *baseline*.

For detection, G_{ed} is implemented by using 5 convolutional layers (except last one, each is followed by batch-norm and Relu). The last conv. layer is followed by a soft-max, thus yielding us EM between 0 and 1. To follow the input image size, we add an up-sampling layer on top of the Error map EM .

For replacement, G_{elr} is implemented through compression block (compresses to $1/64$ of the input resolution) and decompression block (decompresses to $1/4$ of input resolution). These are essentially residual blocks with parameterized skip connection between symmetric layers in decompression and compression blocks respectively.

For additional implementation details of detection and replacement modules, we refer the reader to Section 3.2 of [54].

3.4.3 Training

Using an L1 loss between ground truth and predicted output S' , we optimize using an Adam solver [80], with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rate starting from 10^{-3} is decreased to 3×10^{-4} at 12, 10^{-4} at 18, 3×10^{-5} at 24 and finally 10^{-5} at 28 epochs. We continue training until 32 epochs. Each epoch consists of 500 batch iterations and each batch consists of 16 training samples where each sample is of size 1024×1024 .

Method		Austin	Chicago	Kitsap Co.	West Tyrol	Vienna	Overall
FCN-MLP [105]	IoU	61.20	61.30	51.50	57.95	72.13	64.67
	Acc.	94.20	93.43	98.92	96.66	91.87	94.42
Multi-Task [15]	IOU	76.76	67.06	73.30	66.91	76.68	73.00
	Acc.	93.21	99.25	97.84	91.71	96.61	95.73
Resnet50FCN	IoU	88.12	81.21	83.62	88.15	87.07	86.46
	Acc.	97.02	93.04	99.3	98.28	94.65	96.46
ResnetDR	IOU	89.42	83.56	84.57	89.07	88.30	87.90
	Acc.	97.37	94.03	99.34	98.42	95.12	96.87

Table 3.1 – Evaluation results on validation set

Method		Bellin.	Blooming.	Inns.	S.Fo	Tyrol-E	Overall
FCN-MLP [105]	IoU	56.11	50.40	61.03	61.38	62.51	59.31
	Acc.	95.37	95.27	95.37	87.00	96.61	93.93
Resnet50FCN	IoU	63.34	63.20	76.07	74.91	77.58	72.07
	Acc.	95.90	96.61	97.18	91.67	98.01	95.78
ResnetDR	IOU	64.27	65.85	77.10	75.86	78.68	73.30
	Acc.	95.99	96.52	97.30	92.01	98.11	95.99

Table 3.2 – Evaluation results on test set.

3.5 Experimental Results and Discussions

3.5.1 Quantitative Results

We show the quantitative results in Table 3.1 and Table 3.2 on both the validation and test dataset respectively, where we compare our method (ResnetDR) with the baseline (Resnet 50FCN) and other previous best performing results namely, MLP [105] and Multi-Task Loss [15]. In both tables, we report the Intersection over Union (IoU) and Accuracy score (Acc.) of correct pixels in the segmentation mask ¹. As shown in Table 3.1, we outperformed the previous best method [15] on the validation set by a margin of 14.90% on IoU. While on our own baseline Resnet50FCN, we improve by a margin of 1.44%. On the test set too, we outperformed [105] by 14% and our own baseline by 1.22%.

3.5.2 Qualitative Results

For qualitative analysis, we report our results in Figure 3.3 where green patch represents predictions for each model, red represents ground truth while yellow represents the overlay of ground truth and predictions. We observe that while a major improvement is seen from FCN-MLP to Resnet50FCN, it still is not able to perfectly correct the *blobby* effects. ResnetDR improves upon these blobby effects by not only refining the boundaries of the segmentations but also regularizing them to better reflect structure of building footprints. For *e.g.*in Figure 3.3, in the Chicago image, we note that for the overhead tunnel (bigger circle), ResnetDR yields a more structured output in the form of parallel edges throughout the length of the tunnel. Similarly in case of Tyrol image, in the smaller circle in the middle, predictions overshooting outside the roofs of the houses for Resnet50FCN are constrained in case of ResnetDR to follow the edges of the roof. Finally we see in Vienna image, in the middle circle, for the rooftops of the stands of the playground, ResnetDR enforces some structure by predicting a more rectangular pattern following the building edges.

All these cases proves that our technique learns the underlying structure of building

¹Due to the unavailability of the results on test-set by [15], we haven't reported them in Table 3.2

footprints in the form of regularized and well-defined shapes with straight edges. Hence, it learns to predict error maps that somewhat refine these initial segmentation masks to enforce these rules governing building footprint structures.

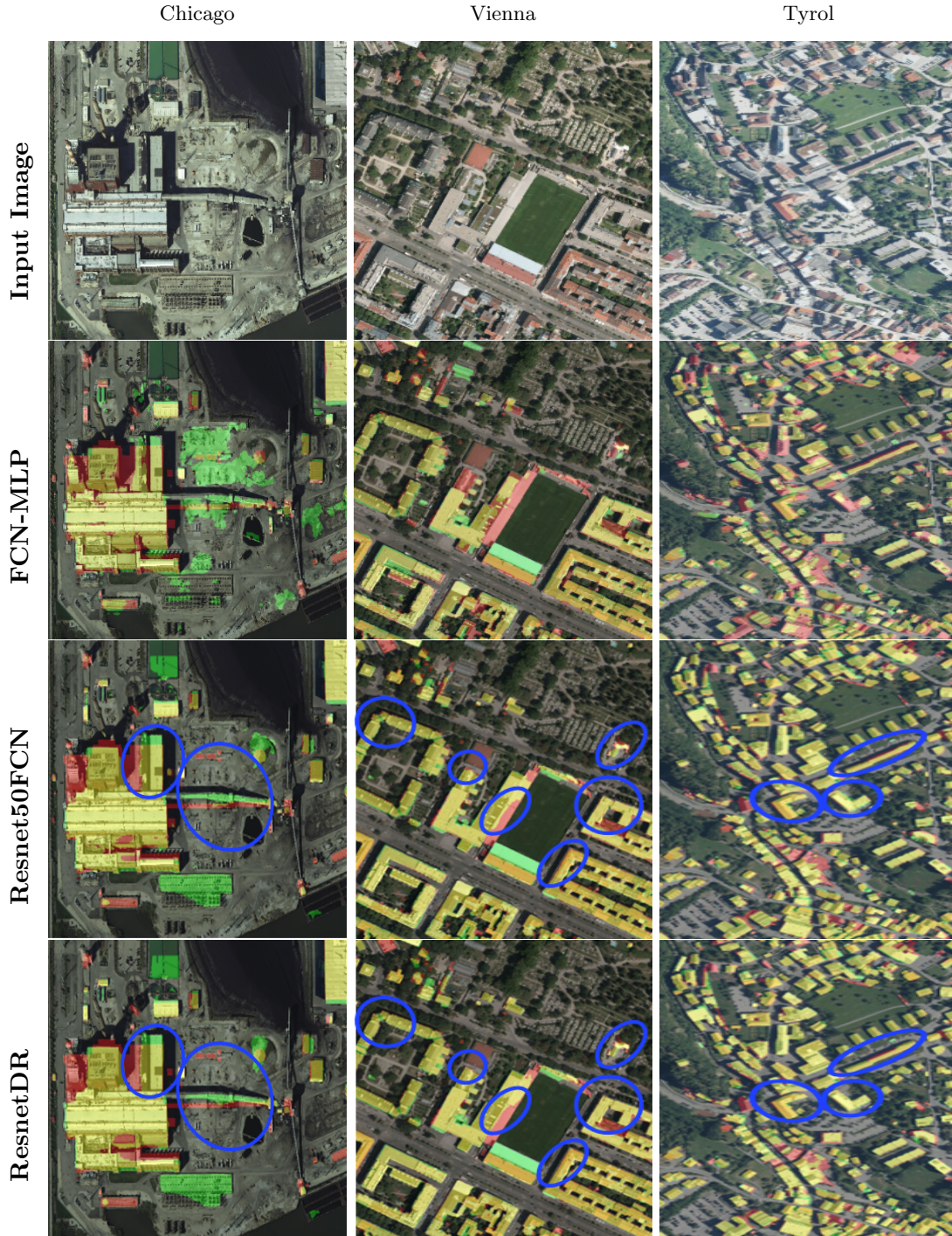


Figure 3.3 – *Qualitative Results*. The ground truth and predictions are overlaid on top of input image in the last three rows. Red segments are the ground truth, green segments are the predictions and yellow segments represent the overlap of ground truth and predictions. The blue circles highlight the important regions where some significant changes have been found as reported in Section 3.5.2

3.6 Conclusions

We present a novel technique for structured semantic segmentation of high resolution satellite imagery. To this end, we propose to learn the joint space of input-output variables. Subsequently, our method enforces structure in the form of predicting a much more regularized building footprint and hence, resolves to a large extent the problem of blobby effects as reported in the past methods. We compare our technique with other state-of-the-art methods on the benchmark Inria aerial image labeling dataset, where we perform at par with other state of the art methods.

Chapter 4

Cloud-GAN: Cloud removal for Sentinel-2 Imagery using a cyclic consistent Generative Adversarial Network

In the previous two chapters, we saw how aerial scene classification and building footprint detection is important for remote sensing image analysis. However, it is often the case that presence of clouds in satellite images makes these computer vision tasks near to impossible, needing an additional image pre-processing at the preliminary level. Cloud cover is a serious impediment in land-surface analysis of Remote-Sensing images causing either complete loss of information due to thick clouds or blurry effects and contrast-reduction in case of semi-transparent thin clouds. While thick clouds require complete pixel replacement with in-painting the underlying surface, thin cloud removal is complex owing to the inter-twining of atmospheric and land-cover information. In this paper, we address this problem and propose a Cloud-GAN model to learn the corresponding mapping between cloudy and its cloud-free images. We employ an adversarial-loss function, that constrains the distribution of generated images to be close enough to the underlying distribution of the non-cloudy images. An additional cycle-consistency-loss is employed to further restrain the generator to predict cloud-free images of the same scene as reflected in the cloudy input image. Our method not only rejects the necessity of any paired training dataset (cloud and its cloud-free counterpart) but also avoids the need of any additional spectral source of information such as Synthetic-Aperture-Radar or Near-Infrared imagery which are cloud penetrable. We demonstrate the efficacy of our technique by training on an openly available and fairly new Sentinel-2 Imagery dataset consisting of real clouds. We validate our model design

by comparing with different variations in the loss function, normalization layer, batch size and the input noise. We lastly compare with other state-of-the-art methods both qualitatively and quantitatively on real as well as synthetic datasets where we witness significant improvement in performance and visual quality, thus validating the competency of our methodology.

4.1 Introduction

Satellites are human eyes in the sky that capture geological, topological and climatic information remotely from hundreds of kilometers away from the Earth. Remote Sensing (RS) images captured by them are pivotal for a wide variety of challenging problems such as recognizing footprints of buildings [105, 153], detecting changes in temporarily apart scenes [99, 152] or semantic segmentation in aerial scenes [3].

Such images are often plagued by films of clouds that partially or completely obstruct the scene. Cloud cover causes significant distortion in visual satellite data by altering either the chrominance or luminance of the satellite image. This can be quite annoying for RS experts performing scene analysis, especially while observing a city like Paris which witnesses cloudy weather for a major part of the year. Thus, it clearly necessitates the requirement for an automatic technique that detects and removes the cloudy regions in a scene and replaces them with the correct background details.

Predicting a scene beneath a cloud is an under-constrained problem. Without any prior information, it is largely quite complex to replace clouds with correct underlying details. A way out has been by using multi-temporal images of the same region as done by [176] through a Multi-Temporal Dictionary Learning. Images from different time-period provide complementary information. However, these methods work only under the assumption that the scene hasn't changed substantially over that period. Authors in [68] use other spectral sources such as Synthetic Aperture Radar (SAR) Imagery, owing to the fact that it can easily penetrate through the clouds. However, SAR imagery has some major drawbacks: a) due to difficulty in its interpretation, and b) having a lower spatial resolution compared to RGB imagery. Additionally, [150, 177] have also studied the thin cloud removal problem in the literature, though, they are based on conventional hand-crafted methods and are limited in terms of performance.

Generative Adversarial Networks (GANs) [59, 72] have gained immense popularity owing to their remarkable capability in modeling the mapping function between input and output domains. Using an adversarial loss, the GAN's can be trained to produce fake images which are indistinguishable from the real images of target domain. Recently, authors in [42] used McGANs to predict cloud-free RGB images as well as cloud masks

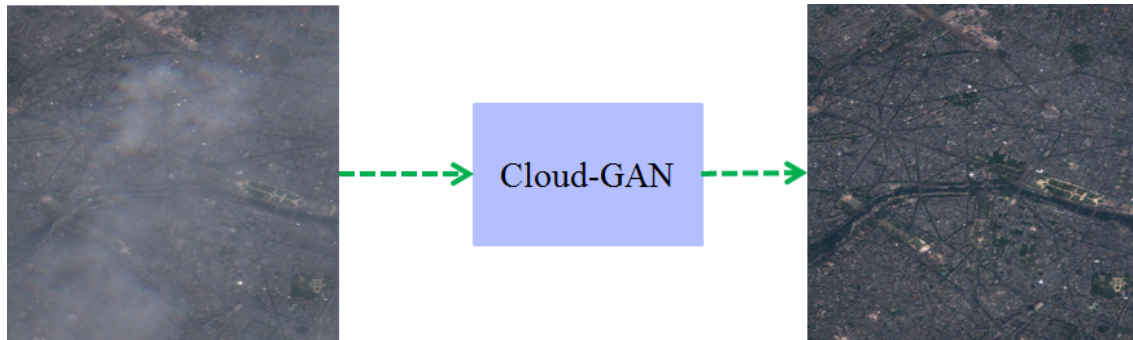


Figure 4.1 – Cloud-GAN can effectively remove clouds from thin cloudy satellite imagery without supervision using ground truth

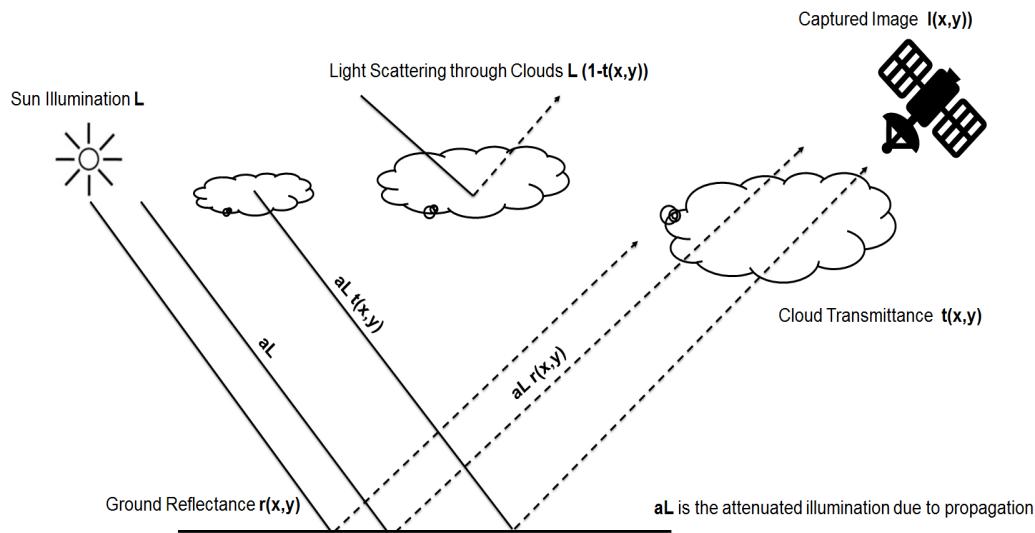


Figure 4.2 – Physical Model of a satellite image capture while transmitting through atmospheric clouds

from the input cloudy image. The authors train their model using pairs of cloud-free and synthetically produced cloudy images. Additionally, they also utilize Near-Infrared (NIR) imagery, which is closer to visible-range and possess partial cloud penetration capabilities. However, there are few associated problems with their methodology. Firstly, synthetically generated clouds are not very realistic and differ significantly from real clouds. Secondly, near infrared images do not provide the complete hidden information behind the clouds as it can only partly penetrate through clouds. Most importantly, composing a large dataset of real clouds and their cloud-free counterparts for training deep-learning models is quite a herculean task.

In this paper, we overcome this hurdle, by improvising upon a novel technique, which after incorporating both the adversarial loss and cycle-consistent loss [188] converts the thin cloudy images to their cloud-free image counterparts. Having a cycle-consistency loss essentially constrains our problem, such that if a satellite image coming from a cloudy

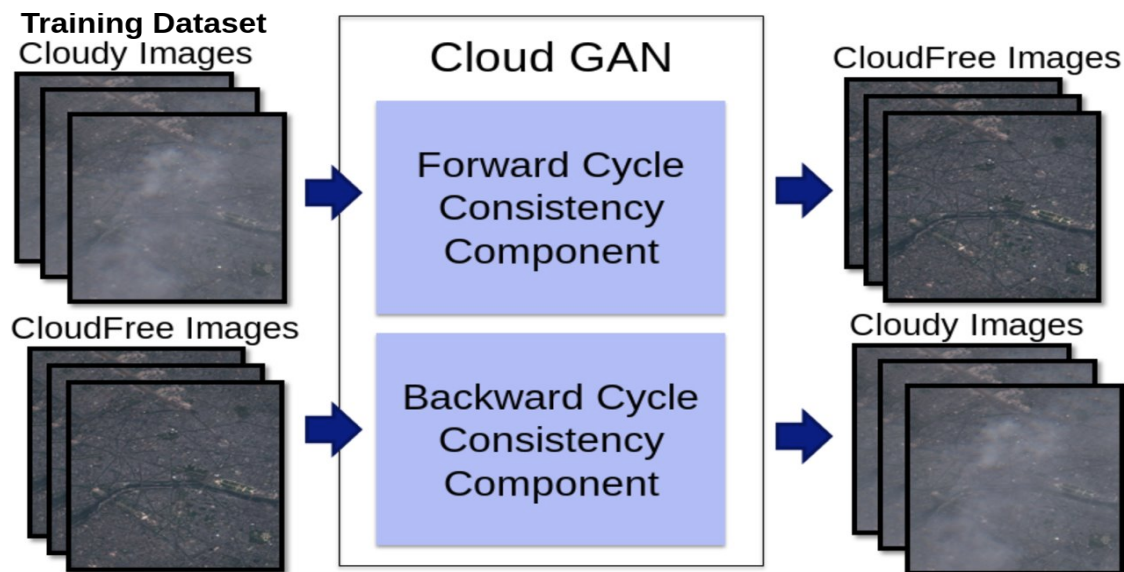


Figure 4.3 – Overall Pipeline of our Cloud-GAN framework

domain is mapped to the cloud-free domain and then transformed back to the cloudy-domain, it should look alike to the original image. An additional advantage of our method is that it liberates us from the requirement of an explicit paired cloudy/cloud-free dataset. Moreover, our methodology doesn't require any sort of cloud-penetration sources of imagery such as either SAR or NIR. We therefore, simply utilize visible range imagery from a fairly new and open source dataset (Sentinel-2) to report impressive qualitative results on both real and synthetically generated datasets clearly showcasing the efficacy of our results. Due to availability of ground-truth, we also report quantitative results for cloud removal of synthetic cloudy images, showcasing a significant improvement compared to other state-of-the-art methods.

Apart from proposing a novel cloud removal technique, we validated the design of our model by conducting an extensive study, discussing the effects of 1) altering the loss function, 2) modifying the type of Normalization layer used within the model, 3) size of input batch and 4) lastly the influence of noise as input to the model. We also draw comparisons with other past openly available state-of-the-art methods, outperforming both on qualitative and quantitative results including considerable improvement in computation-time required for processing a single cloudy-image.

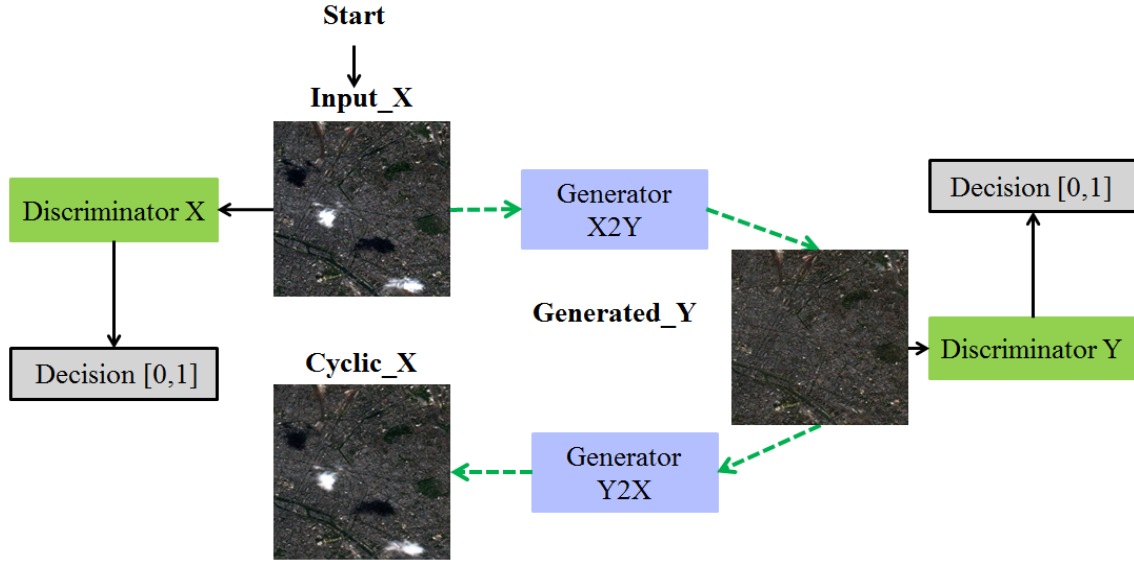
In a nutshell, We:

1. propose a novel technique, to remove thin filmy cloud regions and reconstruct effectively the underlying background in Sentinel-2 imagery.
2. draw extensive comparisons in terms of loss functions, batch-sizes, network-normalization as well as influence of input noise in the form of dropouts to validate our Cloudy-GAN

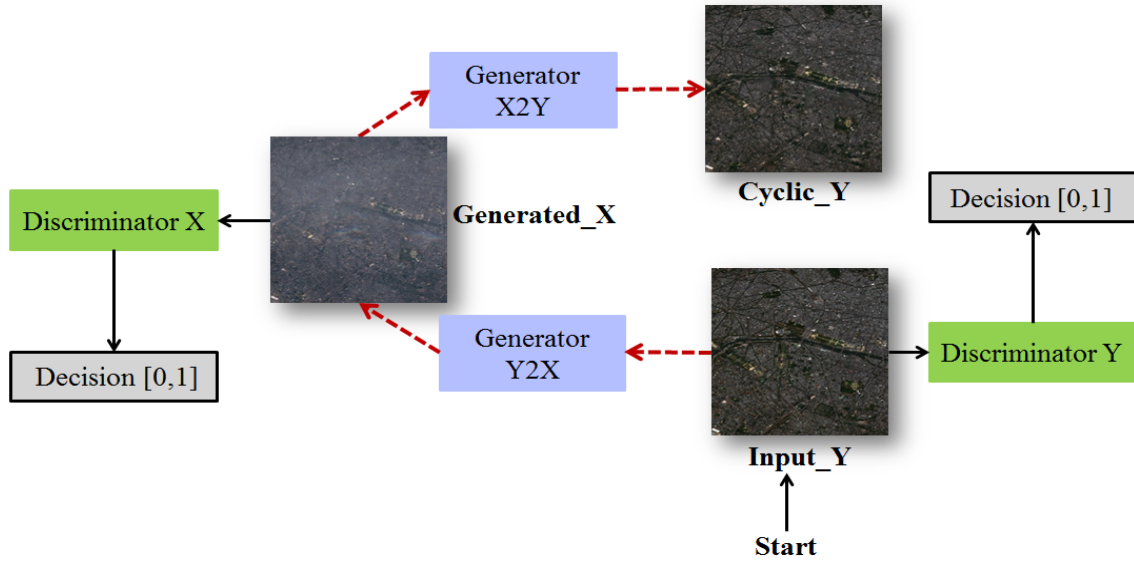
model design.

3. evaluate our proposed method against the past state-of-the-art methodologies, where we outperform them both in terms of (a) qualitative results on real and synthetically generated images and (b) quantitative results only synthetic due to availability of ground-truth. We additionally also hold comparisons with them for processing time of a single cloudy-image.
4. highlight the limitations of our current GAN approach and give possible directions for future work in improving their efficacy for this specific problem setting.

The paper is organized as follows. In Section 4.2, we provide a brief outline of past related works. In Section 4.3, we detail physical model of satellite image capturing under atmospheric clouds. We elaborate our proposed methodology in Section 4.4. We provide training details about dataset, network architectures and other implementation details in Section 4.5. We present the validation of our model design along with experimental results and evaluations with other methods in Section 4.6. Finally, conclusions are drawn in Section 4.7, along with future research directions.



(a) Forward cycle consistency loss $x \rightarrow Q(x) \rightarrow R(Q(x)) \approx x$



(b) Backward cycle consistency loss $y \rightarrow R(y) \rightarrow Q(R(y)) \approx y$

Figure 4.4 – Network Architecture: $Generator_{X2Y}$ and $Generator_{Y2X}$ represent mapping function $Q : X \rightarrow Y$ & $R : Y \rightarrow X$ respectively. Discriminator X and Discriminator Y represent D_X and D_Y respectively.

4.2 Related Work

Cloud removal methods can be broadly categorized into two major classes:

1. Multi-model techniques, that remove clouds utilizing underlying information from multi-spectral or multi-temporal input data.
2. Uni-Model techniques, that remove clouds using only single input data source.

In the following subsections, we initially discuss various state-of-the-art cloud removal methodologies falling broadly under the above two categories. Then, we finally illustrate more recent CNN-based approaches including those using GANs in the context of Cloud Removal.

4.2.1 Multi-Model Techniques

While multi-temporal approaches capture sequence of images of the same scene at different instances in time, multi-spectral utilize the sensitivity of wide variety of sensors, each encapsulating information over a specific wavelength.

Multi-temporal approaches [21, 182, 190] have gained immense popularity for range of tasks be it detecting changes in green cover for deforestation, populating explosion in urban landscapes, crisis management in times of earthquakes/tsunamis or cloud removal over a particular regional area. However, a very fundamental assumption for using such techniques is that their wouldn't be a considerable change in the overall scene topography during the time-period over which these images have been captured [60]. Specifically for cloud-removal, multi-temporal approaches rely on coherence both spatially as well as over time for the underlying captured scene. Though this scene might not change substantially, we do see wide variabilities in the cloud structures over an extended period of time, which effectively makes the task of cloud detection/removal quite challenging [166].

Multi-spectral approaches [131, 132, 135, 150] are increasingly being utilized for a range of practical operations like detecting forest fires, estimating chlorophyll content in plant leaves or mapping mineral deposits. They primarily rely on complementary information captured by each spectral-band independently from each other. In other words, while some bands, might be more sensitive for *e.g.* to green cover in plants, the other might be more efficient in penetrating through the clouds as in case of infrared band. The final signal is thus a fusion of all these diverse sets of interpretations which is much more richer in its overall content. However, these are generally more expensive due to the additional sensors used and at the same time, need way more capacity for storage and transmission.

4.2.2 Uni-model Techniques

Removing clouds using a single image is extremely difficult due to the lack of additional information present in the image behind clouds. Most of the past techniques working on single image cloud removal [97, 101, 143, 163] fill the missing voids by either in painting or interpolation using surrounding pixel information, or adopting various noise removal / image enhancement strategies. They incorporate both high level contextual or structural knowledge as well as low level texture or color information to effectively predict the underlying background of these omitted regions. However, absence of a credible information source in the cloud affected regions limit these techniques for ubiquitous use in removing thin clouds.

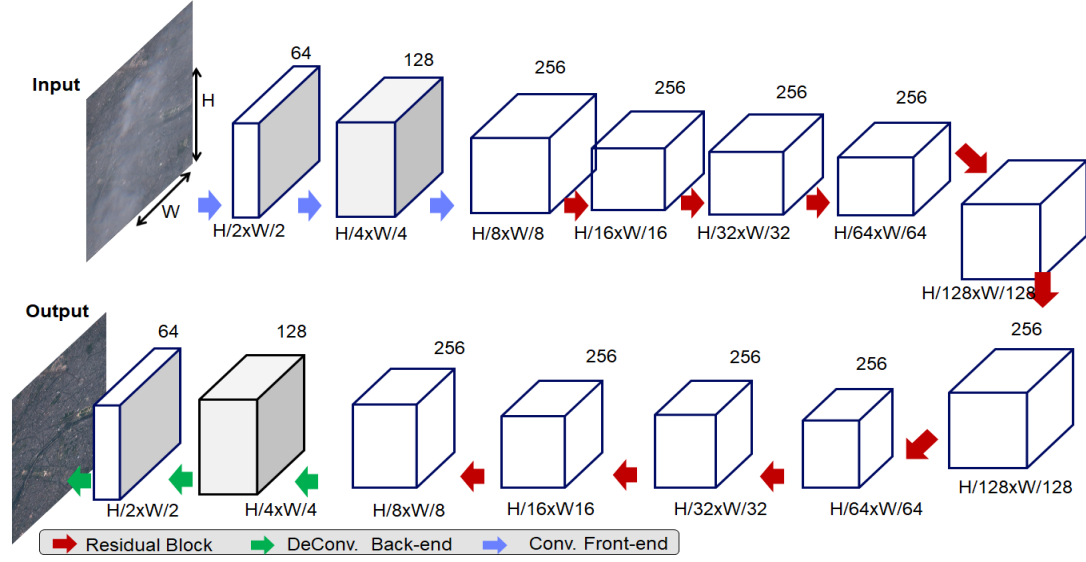
Various noise attenuation methods [62, 96, 185], be it either for fog or haze removal have also aided in removing thin films of clouds. Though a major shortfall of such methods is that they treat clouds with uniform distribution over the image plane. This is quite a strong assumption considering the fact that satellites have a larger region-of-interest covering a wide variety of cloudy patterns in terms of there thickness, structure, overall size or texture. Recent works [91, 147] have constructed distinctive priors to model the cloud and background based on statistical gradient disparities between cloud-free and cloudy images. However, designing such priors require complex understanding of the different features of clouds as well as the background aerial imagery which in turn demands some expert knowledge.

4.2.3 Generative Adversarial Networks

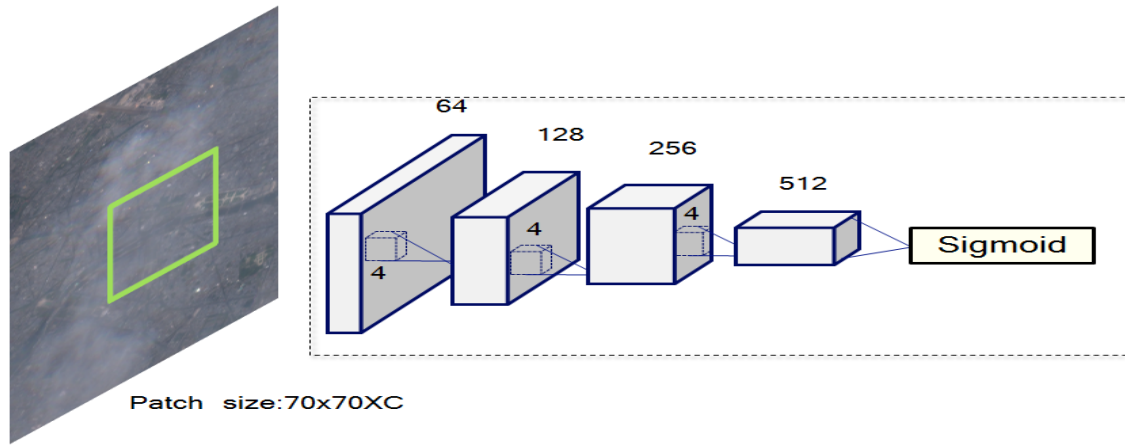
GANs are presently being used for wide varieties of tasks be it vehicle categorization [11], cross-view aerial image synthesis [138], or for hyperspectral image classification [189]. One inherent feature that allows GANs to stand distinctively apart is their ability to effectively model the underlying target distribution. This capability allows them to generate high quality outputs which are visually indistinguishable from the ground-truth dataset. One of the initial works of GAN's [134] improved the stability in training GANs by imbibing a very simplistic CNN model with batch-normalization.

More recent works, either supervised [72] or unsupervised [188] imbibe conditional-GANs (cGANs) [114] for their task of image-to-image translation. They generate outputs conditioned on a specific input image instead of using random noise. Many works [127, 183] have crafted cGANs to solve particularly image restoration tasks or rain and snow removal from natural images. One most related work [42], incorporate RGB as well as NIR images, to remove thin filmy clouds in satellite images. Our approach is fundamentally different from theirs, as firstly we don't require any synthetic paired dataset for training, and secondly

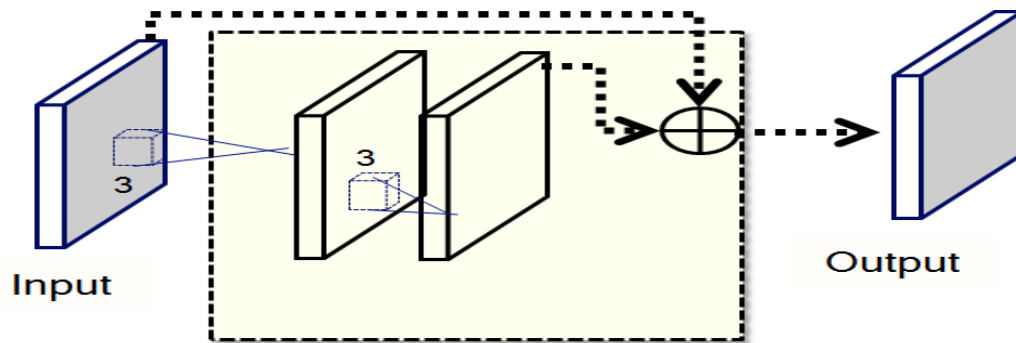
we perform removal solely in a unimodal fashion *i.e.* using single source of RGB image.



(a) Generator architecture



(b) Discriminator architecture



(c) Residual block

Figure 4.5 – Generator and Discriminator architectures including residual blocks.

4.3 Physical Cloud Model

We herein describe the physical cloud model [95] which has been often used for defining prior models for both the overlaying cloud and the background image in various past haze removal techniques [92, 132].

As shown in Fig. 4.2, image signal $I(x, y)$ (x, y representing coordinates of a particular pixel in the image I) goes through various levels of atmospheric obfuscations before being captured by the satellite. It starts from illumination L radiating from the sun, which after traveling through the atmosphere gets attenuated by a factor " a ". Further on, it reaches the ground, where it gets reflected, resulting in signal $a * r(x, y) * L$. This signal then makes its way, either directly to the satellite, or gets obstructed by clouds with a transparency $t(x, y)$. Some part of the illumination gets scattered from the cloud itself without reaching the ground, resulting in $L(1 - t(x, y))$. Hence the final cloudy image degradation model can be described for regions within thin films of clouds with the equation:

$$I(x, y) = ar(x, y)t(x, y)L + L(1 - t(x, y)) \quad (4.1)$$

Cloud removal can be defined as the process of completely obliterating the second term in equation 4.1, which is responsible for reflection from the clouds, while at the same time nullifying the impact of transparency $t(x, y)$ in the first term from the light getting reflected from the ground.

Instead of modeling both these terms using prior functions as done in the past [147], we propose a Cloud-GAN model that effectively learns a mapping from the original cloudy image $I(x, y)$ to a cloud-free version that has been reflected from the ground ($ar(x, y)L$). In the next section, we illustrate in further details, our Cloud-GAN model by first explaining the overall pipeline and then delving deeper into each individual component.

4.4 Proposed Framework

Overall pipeline of the proposed framework is shown in Fig. 4.3. Our Cloud-GAN model is composed of two components, a) forward cycle consistency and b) backward cycle consistency. Both components, while learning on the training dataset, model a mapping function that effectively translates one form of images to another form. While the forward cycle consistency generates cloud-free images from cloudy images, the backward component maps the cloud-free images to cloudy images. We first introduce various terminologies associated with our training set and elaborate the different kinds of Generators and Discriminators incorporated in our model. We, then describe both the forward and backward components in detail.

Assuming cloudy images belong to domain X and cloud-free images to domain Y , we define two mapping functions $Q : X \rightarrow Y$ and $R : Y \rightarrow X$ which are modeled using two generator networks, $Generator_{X2Y}$ and $Generator_{Y2X}$, respectively, as illustrated in Fig. 4.4. Training dataset is composed of $\{x_i\}_{i=1}^N$ and $\{y_j\}_{j=1}^M$ samples, where $x_i \in X$ and $y_j \in Y$ and $p_{data}(x)$ and $p_{data}(y)$ are their data distributions respectively. For simplicity, we will omit use of mathematical format i, j in x_i and y_j and only use x and y in our future notations. As shown in Fig. 4.4, both $Generator_{X2Y}$ and $Generator_{Y2X}$ take as inputs x and y to yield a corresponding output $Q(x)$ and $R(y)$ respectively. Additionally, we have two Discriminator networks, namely Discriminator X (D_X) and Discriminator Y (D_Y) that distinguish between real data (x and y) and adversaries generated by the two Generators i.e. $Q(x)$ and $R(y)$.

Both the Generators and Discriminators effectively play a game; the Generator tries to fool the Discriminator by generating as realistic looking images as possible while the Discriminator tries to identify whether the generated images are real or fake. Hence, in a way the Generators and the Discriminators compete with each other until they reach to a nash equilibrium. At this instant, the Generator is effectively able to match the distribution of generated images ($Q(x)$ & $R(y)$) to the distribution of targeted images ($p_{data}(x)$ & $p_{data}(y)$ respectively).

We specifically utilize Least Square GAN's (LSGAN's) [111], which have shown to generate higher quality images with a much more stable learning process compared to regular GANs. The adversarial objective function is formulated as:

$$\min_{D_Y} L_{LSGAN}(D_Y, X, Y) = E_{y \sim p_{data}(y)} [(D_Y(y) - 1)^2] + E_{x \sim p_{data}(x)} [(D_Y(Q(x)))^2] \quad (4.2)$$

$$\min_Q L_{LSGAN}(Q, X, Y) = E_{x \sim p_{data}(x)} [(D_Y(Q(x)) - 1)^2] \quad (4.3)$$

As shown in equation 4.2, D_Y , having a softmax at its end layer, is effectively trained to yield outputs close to 1 for real images (first term) and 0 for fake images generated by the Generator (second term). On the other hand, the loss function for Q , learns to yield outputs that can trick the Discriminator to reproduce value close to 1; thus asserting that the generated output is quite close to real image.

Similarly, for Discriminator D_X and mapping function R , we have objectives given by:

$$\min_{D_X} L_{LSGAN}(D_X, Y, X) = E_{x \sim p_{data}(x)} [(D_X(x) - 1)^2] + E_{y \sim p_{data}(y)} [(D_X(R(y)))^2] \quad (4.4)$$

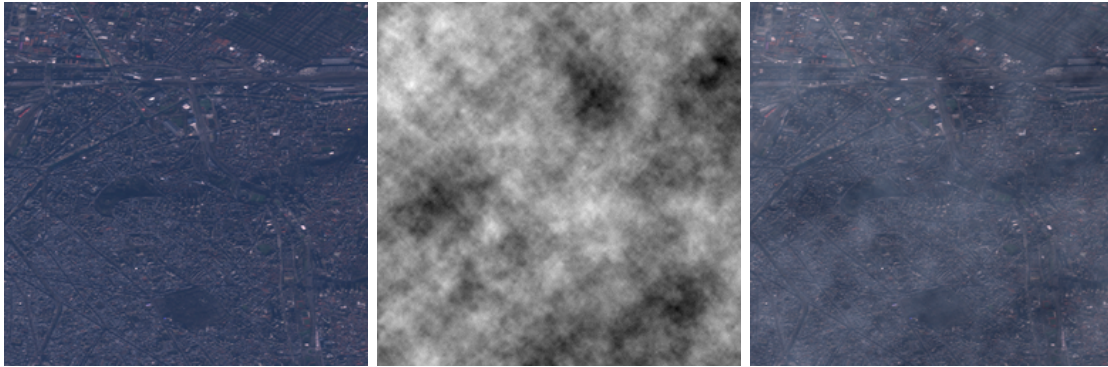


Figure 4.6 – Synthetic Dataset generation using Perlin noise (a) Reference Image, (b) Cloud Mask (c) Synthetically generated image

$$\min_R L_{LSGAN}(R, Y, X) = E_{y \sim p_{data}(y)} [(D_X(R(y)) - 1)^2] \quad (4.5)$$

However, in our kind of problem setting, using an adversarial objective alone is quite under-constrained as the same cloudy image can be mapped to any random permutation of non-cloudy images. Thus, we additionally use a cycle-consistency loss which restrains the Generator to map a given input x_i to a desired output y_i . In Fig. 4.4 (a), this cyclic-loss constrains the final generated output Cyclic X to match Input X *i.e.* $x \rightarrow Q(x) \rightarrow R(Q(x)) \approx x$, which is called as forward consistency loss. Similarly, for backward consistency loss in Fig. 4.4 (b), we have $y \rightarrow R(y) \rightarrow Q(R(y)) \approx y$. We formulate this cycle consistency objective as:

$$\begin{aligned} \min_{Q,R} \mathcal{L}_{cyc}(Q, R) = & E_{x \sim p_{data}(x)} [\| (R(Q(x)) - x \|_1] + \\ & E_{y \sim p_{data}(y)} [\| (Q(R(y)) - y \|_1] \end{aligned} \quad (4.6)$$

Combining the Generator objectives from equations 4.3, 4.5 and the above mentioned cyclic-consistency loss, our final Generator objective can be formulated as:

$$\begin{aligned} \min_{Q,R} \mathcal{L}_{Gen}(Q, R) = & \mathcal{L}_{LSGAN}(Q, X, Y) + \\ & \mathcal{L}_{LSGAN}(R, Y, X) + \lambda \mathcal{L}_{cyc}(Q, R) \end{aligned} \quad (4.7)$$

where λ is a regularizing factor that weights the cyclic term with respect to Generative term of the adversarial objective function and is set to its default value 10.

4.5 Training and Implementation

4.5.1 Dataset

We build our training and test dataset by downloading high resolution Sentinel-2 images from Copernicus Open Access Hub (<https://scihub.copernicus.eu/>). Sentinel-2 is a constellation of two Earth observation satellites (Sentinel-2A and Sentinel-2B) launched by European Space Agency, and is part of the European Commission’s ambitious Copernicus Earth observation program. This program is the most exhaustive Earth Observation program ever created, providing observations related to land, ocean, and atmosphere by revisiting each location on the Earth’s surface every 5 days. Sentinel-2 data is quite suitable for various land mapping applications be it urban planning, forestry or agriculture.

The Earth images so captured under Sentinel-2 mission have a resolution ranging between 10-60 meters per pixel while covering a field of view of 290 km in 13 bands including visible and infrared. These images are duly processed to Level-1C, *i.e.* they are ortho-rectified, map-projected images containing top-of-atmosphere reflectance data. Each of the 13 spectral bands of the multi-spectral dataset are stored as separate image. For our experiments, we choose images only from visible bands *i.e.* Blue (B2), Green (B3), Red (B4) all of which have 10 meters of spatial resolution.

We utilized Medusa toolbox from Onera [121] to download a stack of Sentinel-2 images from Jan 2015 to August 2017 over an urban region of Paris whose geo-coordinates are provided as a geojson file. The toolbox provides with an efficient python script using which we can define the % of cloud-cover in each downloaded image. We select 0-5% cloud cover for cloud-free images, while for cloudy images we chose a range anywhere between 10 to 100%. We download images over Paris as it is easy to get quite a range of cloudy cover above it. Out of the total downloaded images, we choose 20 cloudy and 13 cloudless images. We then extract 512×512 patches from these images using a running window over each image. After filtering the unwanted ones, we extract a total of 1677 training patches for each cloud and cloud-free dataset while for testing we had 837 patches. For computational efficiency and GPU memory limitation, we resize them to 256×256 at the time of training.

Although we use real cloudy images for training, we cannot perform quantitative evaluation on them due to the absence of a freely available paired cloudy cloud-free image database. Hence, we constructed our own dataset of synthetic images in order to further validate the performance of our network by comparing it to other methods. Previous studies [33, 42] have utilized Perlin noise [129] to generate clouds due to its characteristic feature of procedural texture generation. We first generate thin films of cloud masks by using Perlin noise and then overlay them onto the reference image using alpha blending as shown in Fig. 4.6. The reference image (Fig. 4.6(a)) is blended using the cloud mask

Table 4.1 – Comparisons for variability in model design in terms of loss function, normalization layer, batch size, and input noise over Synthetic dataset

Type	Euclidean Distance
CloudGAN without LSGAN (standard GAN)	10710.57
CloudGAN without Cyclic-consistency loss ($\lambda=0$)	5023.56
CloudGAN with BatchNorm	5500.36
CloudGAN with batchsize=1	10659.61
CloudGAN with noise (Dropout)	9447.06
Cloud-GAN (Our model)	4767.94

Table 4.2 – Quantitative Results showcasing Average PSNR, SSIM and RMSE scores on Synthetic Dataset

Methods	PSNR	SSIM	RMSE
Jobson [74, 124]	9.79	0.26	143.09
He [62]	16.27	0.64	68.71
Tarel [155, 156]	17.35	0.79	70.85
Berman [12, 13]	15.95	0.66	59.98
Ren [141]	19.89	0.75	44.84
Cloud-GAN	27.56	0.80	18.62

(Fig. 4.6(b)) to yield our synthetically generated image (Fig. 4.6(c)).

4.5.2 Network Architectures

We imbibe the architecture and naming convention similar to what have been used by [188]. Our Generator architecture is in the form of an encoder-decoder layer where we first down-sample the input to a compressed representation and then up-sample it to the same size as the input. As shown in Fig. 4.5 (a), the Generator architecture starts with down-sampling using a 7×7 Convolution + Instance Normalization (Instance-Norm) [161] + Relu layer having stride 1 resulting in 64 output feature maps. We will show the efficacy of using Instance-Norm instead of Batch-Norm [70] in Section 4.6. Not using Normalization in the Generator layers results in inferior results. The next two layers are 3×3 Conv + Instance-Norm + Relu layers with 128 and 256 filters and stride 2 respectively. This is followed by 9 residual blocks (as shown in Fig. 4.5 (b)), each containing 3×3 convolution filters resulting in same output layers for both. As detailed in Section 4.6, we additionally try to test the addition of noise to the Generator architecture in the form of dropouts inside the residual blocks. Next, we perform up-sampling using 2 transposed convolution layers along with Instance-Norm and Relu having 128 and 64 filters and stride 2 respectively. The last layer is a 7×7 convolution layer that yields final 3-channel output from the generator. Note that before each convolution, we use reflection padding on the boundaries for artifact correction.

Table 4.3 – Computation Times for different methods

Methods	Time (seconds)
Jobson [74, 124]	0.5813
He [62]	0.3722
Berman [12, 13]	0.1847
Tarel [155, 156]	0.4733
Ren [141] ¹	0.5140
Cloud-GAN	0.045

The Discriminator architecture on the other hand resembles a 70×70 PatchGAN [72] architecture, classifying 70×70 patches as real or fake data. Compared with a discriminator working on full-scale images, ours can effectively be applied to any input size image with significantly lesser number of parameters. The architecture starts with a layer with 4×4 Convolution + Leaky-Relu of slope 0.2 which is applied to the input image to yield a 64-channel feature map. This is followed by 3 convolution + InstanceNorm + Relu layers of filter size 128, 256 and 512 respectively having a stride 2. In the end, we apply a convolution layer resulting in a 1-dimensional feature vector followed by a Sigmoid function.

4.5.3 Implementation Details

Our network is trained using a 12 Gb NVIDIA Titan-X GPU on an Intel Xeon e7 core i7 machine using Pytorch [125] as the deep learning framework. Initialization of weights was done through a Gaussian distribution with mean 0 and standard deviation 0.02. Optimization was carried out using ADAM [80], with a batch size of 4 and $\lambda = 10$ as default value for all experiments. We perform training from scratch using a learning rate of 0.0002 up-to 200 epochs. The learning rate was kept constant for the first 100 epochs after which it linearly decays to zero until the last epoch. Also, as illustrated in [188] model oscillations are avoided by using a history of generated images (50) rather than only one.

4.6 Results and Evaluation

We validate our model by comparing it with other possible variations in its input batch-size, normalization layers, noise as input, replacing LSGAN with a standard-GAN or removing cyclic-consistency loss from overall objective function. Consequently, we test this verified architecture against different state-of-the-art methods both on real and synthetic images. While for real dataset, we show only qualitative results owing to lack of ground truth, for synthetic images we compare both qualitatively as well quantitatively with other methods.

¹Run on a separate Intel Core i7 windows machine with NVIDIA GeForce MX150 Graphics card (2GB memory), though we assume the performance to be much faster if run on a Titan-X GPU as used in our case.

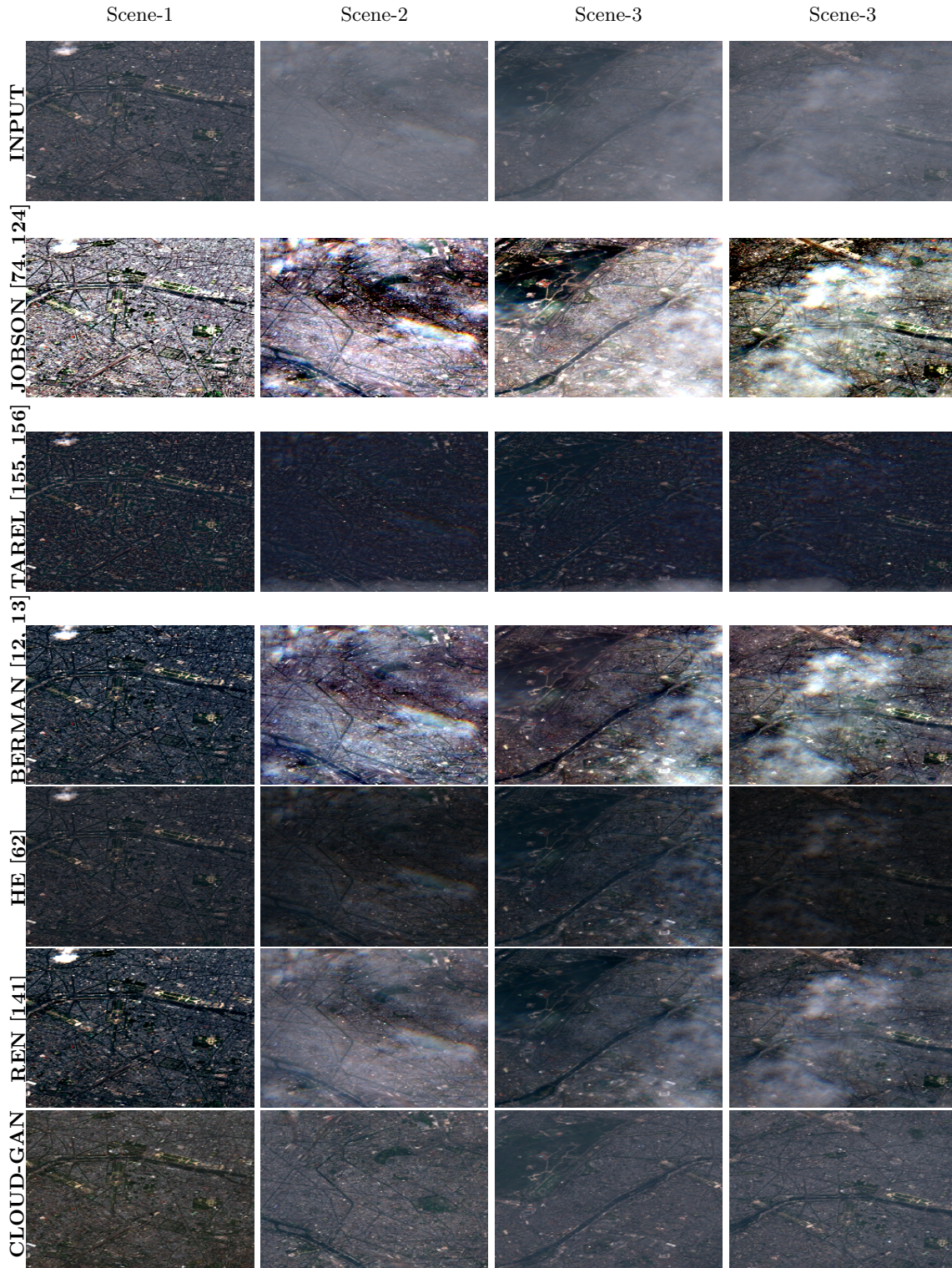


Figure 4.7 – *Qualitative Results* on Real Cloud Dataset. The first row is the input cloudy image while last row is the output from our Cloud-GAN model. All other rows represents results from other state-of-the-art methods.

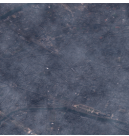
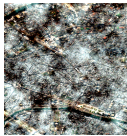
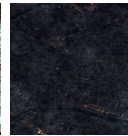

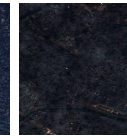
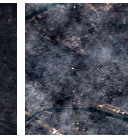



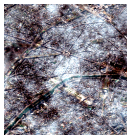
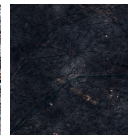
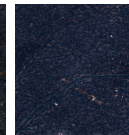

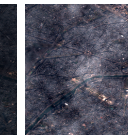


	Input	Jobson [74, 124]	He [62]	Tarel [155, 156]	Berman [12, 13]	Ren [141]	Cloud-GAN	Ground- Truth
Synthetic Scene-1								
	24.01 dB	10.15 dB	14.41 dB	16.45 dB	15.20 dB	19.18 dB	27.79 dB	←PSNR
Synthetic Scene-2								
	23.76 dB	10.14 dB	15.04 dB	16.66 dB	15.82 dB	20.15 dB	28.34 dB	←PSNR

Figure 4.8 – Comparison of various cloud removal techniques for two synthetically generated scenes. For quantitative evaluation, we show PSNR values which are computed for each image starting from Input and all other methods against the Ground-Truth.

Lastly, we also hold comparisons for computation time required to process an image for all the methods.

4.6.1 Validation of model design

As described in Section 4.4 and 4.5, in our Cloud-GAN model, we employ LSGAN together with a Cyclic-loss as the loss function, Instance-Norm as normalization layer within the network, batch size is set to 4 and inputs for $Generator_{X2Y}$ and $Generator_{Y2X}$ are either Cloudy or Cloud-free images without any random noise. In order to showcase the efficacy of our model, we compare it with 5 possible design variations in the end loss function, normalization layer, batch size, addition of noise and removal of cyclic-loss ($\lambda=0$).

As illustrated in Table 4.1, we use euclidean distance to evaluate cloud-removal results from different design variations. Clearly, a lower value of Euclidean distance indicates better performance. All the comparisons are drawn with our self generated synthetic dataset where we have ground-truths available for each corresponding cloud-free image generated by the models.

As can be seen, our model using LSGAN results in a higher performance compared to standard GAN [59]. This is mainly because LSGAN, compared with standard GAN, do not have a log-loss function for the discriminator. This effectively results in penalizing those data-points which are far-off in distance from the decision boundary, thus yielding smooth and non-saturating gradients for discriminator.

Removing cyclic-loss term by setting $\lambda=0$ also further degrades the performance of our network. This implies that cyclic-loss is quintessential for mapping input cloudy /

non-cloudy image to its corresponding cloud-free / non-cloudy image in both the forward-consistency and backward-consistency components. The reason is quite evident as without the cyclic term, the network maps the input image to any randomly chosen sample from the target space which results in poor performance.

Next we see that instance-norm (our model) gives better performance compared to when using batch-norm (normalizing over entire batch). This is primarily due to satellite images present in a batch have wide-scale illumination variation and thus, they need to be effectively normalized independently than in a batch for stable training.

We also see that our model with batch size = 4 yields better performance than one trained with batch size=1. This can be reasoned as the Generator architecture tend to have larger variability in samples with a larger batch size than a smaller one when learning a mapping from cloudy / cloud-free images to cloud-free/cloudy images. Thus effectively at each gradient descent step, it tries to minimize the adversarial loss by generating wider range of cloudy/cloud-free image. Note that batch size = 4 is the maximum we could take due to limitation of our GPU memory.

Lastly, to avoid smoothness in the predicted outputs (Fig. 4.9) and introduce some stochasticity in the output predictions, we tried to add random noise using dropouts in the generator layers. However, what we see is an overall drop in the performance as introducing such noise also impacts other images with clean outputs and leads to deterioration of structures in those images.

Hence, finally we conclude that our model design with LSGAN and cyclic loss as loss function, instance-norm, batch-size=4 and without using any noise performs the best among all possible design variations that we test. Next, we use this model for comparisons with other state-of-the-art methods.

4.6.2 Comparison with state-of-the-art methods

We herein present comparisons of results obtained using our Cloud-GAN model with other openly available methods such as He [62], Jobson [74, 124], Tarel [155, 156], Ren [141] and Berman [12, 13] both on real-cloudy images and synthetic-cloudy images. We first demonstrate qualitative results over real-cloudy images and then draw comparisons both quantitatively and qualitatively over synthetically generated cloudy images. Note that we do not provide comparison with some more recent works [42, 132, 147], due to the unavailability of their code and dataset.

Comparison with Real-Images

Fig. 4.7 shows comparisons for real-cloudy inputs with different variations in cloud cover. While our method (row VII) is able to remove both thick (to some extent) as well as thin cloudy films, Jobson, Berman and Ren in rows II, V and VI respectively are unable to completely recover thick cloudy regions. This is owing to the fact that dense saturated regions cannot be solely treated with image enhancement based methods and require additional image in-painting methods [97, 101] to recover the background. In case of Jobson, we additionally witness over-enhanced contrasts which isn't desirable while removing clouds. For He and Tarel (rows III and IV respectively), we see that they can remove these dense cloud-covers to large extent, however, they suffer from diminished contrast and dark regions.

Our Cloud-GAN model (row VII), without learning using any kind of Cloud-Free cloudy pair images, efficiently removes thin clouds and small thick-cloudy regions spread throughout a scene. More interestingly, it effectively detects small cloudy patches and replaces them with the underlying ground details, as depicted in scene-1. At the same time, it can retain finer details like patches of urban settlements, rivers, fields (scene-3) while getting rid of the thin cloudy film. In some cases such as scene-1, the generated image from our method is more natural and visually much more pleasing than the original image which is another byproduct feature of using Cloud-GAN. Additionally, we neither witness any over-enhancement/diminishing contrast problem as reported in other methods, nor any kind of dark patches or global illumination degradation.

Comparisons with Synthetic-Images

We cannot report any quantitative results on the real dataset since we lack paired cloudy-cloud-free images. However, to compensate that, we report results on synthetic scenes, which were composed by addition of Perlin noise to cloud free images as elaborated in Section 4.5.

In Fig. 4.8 we show results with different cloud removal methods including ours over two synthetically generated cloudy scenes together with their corresponding Peak Signal-to-Noise-Ratio (PSNR) values. For the last column-VIII, being a ground-truth image, we do not report its PSNR value. We see that in terms of PSNR values, our model performs considerably better than even the best performing method which is Ren (column VI). On visually inspecting the results, we find similar patterns as found in real-dataset. We typically see that Jobson, Berman and Ren (Columns II, V and VI respectively) are unable to remove the cloudy cover completely. While Jobson overly enhances the contrast (resulting in lower PSNR), He and Berman (Column III, IV and V resp.) compress the contrast.

Next, we report quantitative results in Table 4.2 using three popular metrics namely

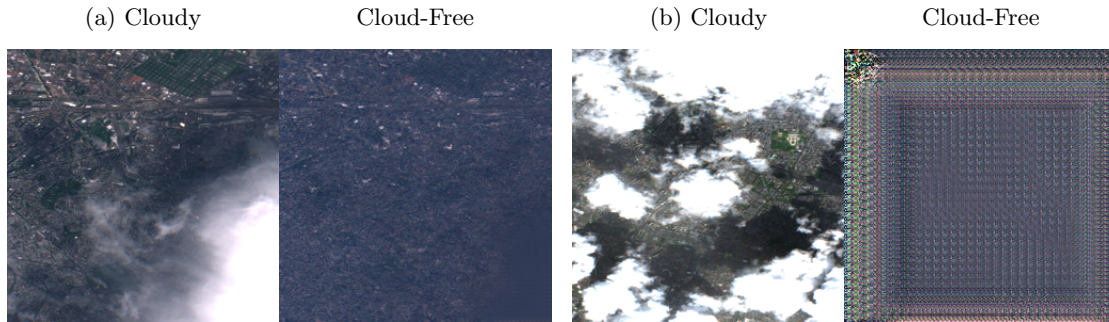


Figure 4.9 – Failure Cases: Over-smoothing or completely fail to produce images for overly clouded images

PSNR, structural similarity (SSIM) [169] and Root Mean Square Error (RMSE) to compare difference in variability’s over all synthetic scenes. For PSNR value, we find that on an average all methods perform approximately 8-18 points below our model. Specifically for Jobson, we see very-low PSNR values due to extreme over-enhancement of contrast which hugely deteriorates the overall perceptual quality of the image. For SSIM index, which in a way measures both structural information and perceptual quality, we see that Jobson, He and Bermen perform quite poorly compared to our method due to extreme changes in the luminance and contrast of their corresponding outputs. For Tarel and Ren, the results are quite comparable to ours. Similarly for RMSE, we notice that Jobson has the maximum absolute error, due to again per-pixel large changes between the target image and its output. Overall our method outperforms the best one (Ren) by a margin of 26 points.

4.6.3 Computation-time

We additionally show comparisons for computational time taken to remove clouds over a single satellite image in table 4.3. On an average our method is 10 times faster compared to other past methods. While He [62] takes 0.37 seconds, for Tarel [155, 156] it takes 0.47 seconds and for Jobson [74, 124] it is 0.58 seconds. The fastest of them all Berman [12, 13] is still 4 times slower than our Cloud-GAN. Note that due to platform constraints for their code, Ren [141] was run on a separate Intel Core i7 windows machine with NVIDIA GeForce MX150 Graphics card (2GB memory). Though running on this configuration their algorithm takes 0.51 seconds, we assume the performance to be much faster if run on a Titan-X GPU as used in our case.

4.6.4 Results with Dense cloud cover

We show some special instances of thick clouds (Fig. 4.9) where our model fails to yield credible results. In Fig. 4.9 (a), we see that the model generates an overly-smoothed image

when the clouds are too opaque. In Fig. 4.9 (b), the model fails completely to produce an image as the clouds have occupied most of the visible area. One of the plausible reasons is that the network finds no closest sample in the target dataset associate with such thick cloudy cover and hence, predicts a spatially smooth region under the cloud or some random noise when the cloud cover is over large parts of the image.

A possible way to address this is by the addition of an extra source such as SAR imagery which can penetrate through these clouds and give us information about the underlying ground details. Note that we checked with NIR imagery too, however, they are themselves partially plagued by cloud cover and could not help much. Moreover, curating only those regions which are present beneath a cloud and leaving other non-cloudy regions as such is another future area of research. This problem can possibly be addressed by using some form of masking mechanism as has been utilized for refining building segmentation results in one of the more recent works [153].

4.7 Conclusions and Discussions

We have proposed a novel technique to remove thin filmy clouds from Sentinel-2 imagery. Our cloud removal technique is employed without using any explicit dataset of paired Cloudy/Cloud-Free images as performed in the past. In a way, it nullifies the requirement of creating and training on synthetic dataset which is not truly realistic. Our input sources are purely visible range images without any prerequisite for other spectral sources such as NIR, SAR or other cloud-penetration sources. We legitimize the design of our model over wide-range of variabilities in terms of loss function, normalization layers, batch size and input noise. Lastly, we report significant improvement over state-of-the-art methods in both qualitative as well as quantitative results on synthetic images as well qualitative results on real images, validating the efficacy of our results. In terms of computation time too, our model performs considerably better than other past techniques, effectively allowing real time processing of cloudy-images. However, presence of thick clouds do necessitate need of an additional high wavelength imagery to gain some knowledge about the underlying ground information which we leave as a topic of future research. Additionally, it would also be interesting to see how the model performs while testing it on cities other than Paris where the cloud cover is lesser or more as compared to it.

Chapter 5

Deep Tone Mapping for High Dynamic Range Scenes

In the last three chapters, we discussed various applications of deep learning for processing high resolution satellite images. A high spatial resolution in EO data simply implies having smaller area to cover for each pixel, thus effectively being able to distinguish finer details on ground from the imagery. In this chapter, we are going to drift towards effectively processing another set of high resolution images, which is broadly focused on high dynamic range content or popularly termed as HDR images. Contrary to EO data, a high spatial resolution in HDR imagery largely concerns with larger number of pixels per inch irrespective of the captured area in the scene. A computationally fast tone mapping operator (TMO) that can quickly adapt to a wide spectrum of HDR content is quintessential for visualization on varied low dynamic range (LDR) output devices such as movie screens or standard displays. Existing TMOs can successfully tone-map only a limited number of HDR content and require an extensive parameter tuning to yield the best subjective-quality tone-mapped output. In this chapter, we address this problem by proposing a fast, parameter-free and scene-adaptable deep tone mapping operator (DeepTMO) that yields a high-resolution and high-subjective quality tone mapped output. Based on conditional generative adversarial network (cGAN), DeepTMO not only learns to adapt to vast scenic-content (*e.g.* outdoor, indoor, human, structures, etc.) but also tackles the HDR related scene-specific challenges such as contrast and brightness, while preserving the fine-grained details. We explore 4 possible combinations of Generator-Discriminator architectural designs to specifically address some prominent issues in HDR related deep-learning frameworks like blurring, tiling patterns and saturation artifacts. By exploring different influences of scales, loss-functions and normalization layers under a cGAN setting, we conclude with adopting a multi-scale model for our task. To further leverage on the large-scale availability of unlabeled HDR

data, we train our network by generating *targets* using an objective HDR quality metric, namely Tone Mapping Image Quality Index (TMQI). We demonstrate the results both quantitatively and qualitatively, and showcase that our DeepTMO generates high-resolution, high-quality output images over a large spectrum of real-world scenes. Finally, we evaluate the perceived quality of our results by conducting a pair-wise subjective study which confirms the versatility of our method.

5.1 Introduction

Tone mapping is a prerequisite in the high dynamic range (HDR) imaging [58, 100, 123] pipeline to print or render HDR content for low dynamic range displays. With the unprecedented demands of capturing/reproducing scenes in high-resolution and superior quality, HDR technology is growing rapidly [36, 113]. Although HDR display systems have advanced in the last few decades (for *e.g.* Sim2, Dolby Vision, etc), they still necessitate some sort of tone mapping operation because of limited technical capabilities of the materials used in these displays. Additionally, due to high manufacturing costs of the rendering mediums, the absolute majority of displays/screens still has limited dynamic range and relies largely on Tone Mapping Operators (TMOs) for desired top-quality rendering.

Several TMOs have been designed over the last two decades, promising the most faithful representation of real-world luminosity and color gamut for high-quality output. However, in practice, such TMOs are limited to successfully tone map only limited number of HDR images due to their parametric sensitivity [40, 86]. For instance, a TMO capable of mapping a bright daytime scene might not map the dark or evening scenes equally well. In fact, one needs to manually tweak in an extensive parametric space for every new scene in order to achieve the best possible results while using any such TMO. Thus, the entire process of finding the most desirable high-resolution tone-mapped output is not only utterly slow, tedious and quite expensive, but is almost impractical with a large variety of HDR content being generated daily from different capturing devices.

This raises a natural question whether a more *adaptive* tone mapping function can be formulated which can quickly alter itself to wide variability in real-world HDR scenes to reproduce the best subjective quality output without any perceptual damage to its content under a high-resolution rendering. With the recent success of deep learning [82] and wide scale availability of HDR data, it is now quite possible to learn a model with such complex functionalities for effective tone mapping operation.

In this chapter, we propose an end-to-end deep learning (DL) based tone-mapping operator (DeepTMO) for converting any given HDR scene into a tone-mapped LDR output which is of high resolution [1024x2048] and superior subjective quality. Based upon a



Figure 5.1 – Comparison between CNN (encoder-decoder) with L_1 -loss and DeepTMO (single-scale). Insets in row 2 show that DeepTMO yields sharp and high resolution output, whereas the CNN results in blurred outputs.

conditional generative adversarial network (cGAN) [59, 114], the DeepTMO model directly inputs 32-bit *linear* HDR content and reproduces a realistically looking tone-mapped image, aiming to mimic the original HDR content under a limited range [0-255]. DeepTMO is trained to cater a wide range of scenic-content for *e.g.* indoor/outdoor scenes, scenes with structures, human faces, landscapes, dark and noisy scenes, etc.

The motivation for generative adversarial network (GAN) in the DeepTMO design stems from the poor performance of conventional convolutional neural networks (CNNs) explored in past HDR studies [40, 41, 52] where the images resulted with spatially blurred out pixels by using a simple L_1/L_2 loss function. Essentially, such models are not ideal for our tone-mapping problem where the goal is to obtain output images that are: artifact-free, superior-quality and high-resolution. Furthermore, instead of optimizing parameters for a given TMO for one particular scene [29, 100], our objective is to design a model which is *adaptable* to different scenes-types (such as day/night, outdoor/indoor, etc.), thus encompassing all the desired characteristics for those scenes. Altogether, this is difficult for a naive loss-function to satisfy. Moreover, designing such a cost function is quite complex, and needs expert knowledge. Therefore, we overcome this challenge by *learning* an ‘adversarial loss’ that encapsulates all the desired features from all ideal tone-mapped images by using the underlying training data; thereby eradicating the need of manually handcrafting such a loss function.

GANs are capable to generate better quality images compared to the state-of-the-art models, however, there are still some prominent issues such as tiling patterns, local blurring and saturated artifacts (see Fig. 5.6 (a)). To handle these problems in a high-resolution output image, we explore the DeepTMO architectural design by comparing the single-scale and multi-scale variants of both generator and discriminator. We subsequently showcase how a multi-scale version of the generator-discriminator architecture helps in predicting artifact-free tone mapped images, which are both structurally consistent with input HDR and simultaneously preserve fine-grained information recovered from different scales.

The DeepTMO model is effectively a multi-scale architecture having a 2-scale generator and a 2-scale discriminator, both of which are conditioned on the *linear* HDR input. Both generator and discriminator compete with each other. The generator is trying to fool discriminator by producing high subjective quality tone mapped images for the given input HDR, while the discriminator trying to discriminate between real and synthetically generated HDR-LDR image pairs. Our basic discriminator architecture is similar to Patch-GAN [87, 88] which classifies patches over the entire image and averages over all of them to yield the final image score. Similarly our basic generator architecture comprises of an encoder-decoder network where the input HDR is given first to an encoder resulting in a compressed representation which is then passed to the decoder yielding finally a tone



Figure 5.2 – Comparison between CNN (encoder-decoder) with L_p -loss and DeepTMO (single-scale).

mapped image.

To train our model, we accumulate our dataset from freely available HDR image sources. Ideally, the training dataset should be created through a subjective evaluation considering all possible tone mapping operators for all available HDR scenes. However, conducting such a subjective evaluation is highly cumbersome and unfeasible. Thus, it necessitates the requirement of an objective quality assessment metric which can quantify the tone mapping performance of each TMO for any given scene. For our task, we select a well known metric namely Tone Mapped Image Quality Index (TMQI). We first rank 13 widely used TMOs using the TMQI metric for each HDR input. We then select the topmost scoring tone-mapped image as our *target* output.

In a nutshell, we

1. propose a fast, parameter-free DeepTMO, which can generate high-resolution and foremost subjective quality tone-mapped outputs for a large variety of *linear* HDR scenes, including indoor, outdoor, person, structures, day and night/noisy scenes.
2. explore 4 possible cGANs network settings: (a) Single-scale-Generator (Single-G) and Single-scale-Discriminator (Single-D), (b) Multi-scale-Generator (Multi-G) and Single-D, (c) Single-G and Multi-scale-Discriminator (Multi-D), (d) Multi-G and Multi-D, thus discussing the influence of scales and finally adopting a multi-scale generator-discriminator model for our problem.
3. detail the impact of different loss functions and normalization layers while elaborating how each step helps in improving the overall results by tackling different artifacts.

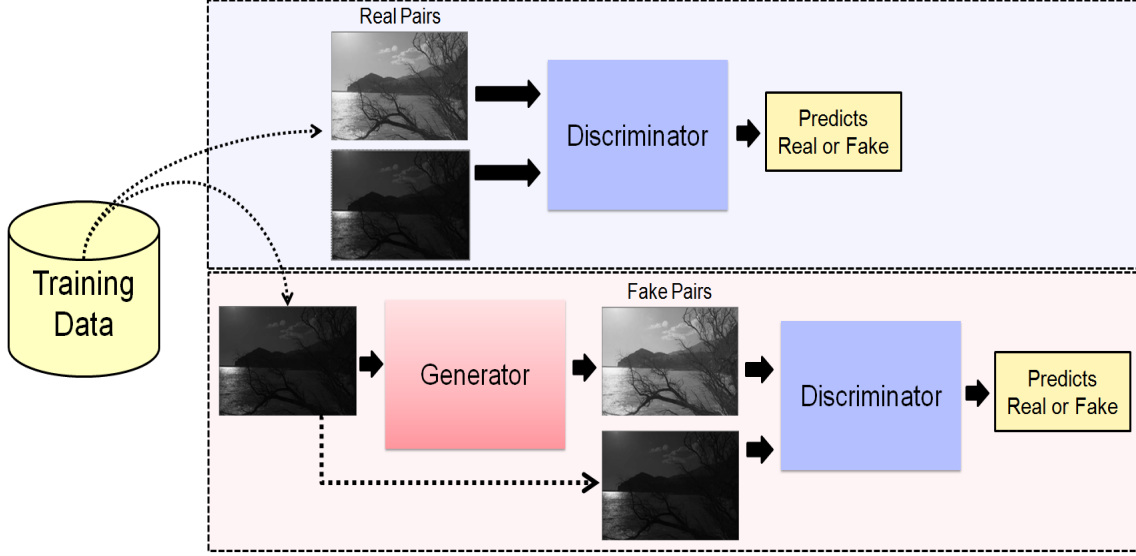


Figure 5.3 – DeepTMO Training Pipeline.

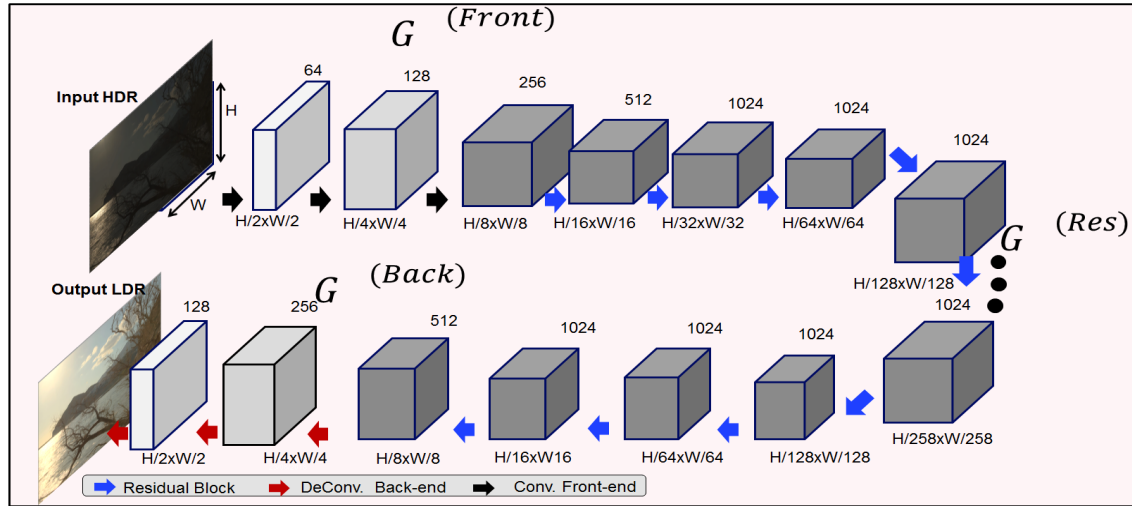
4. provide quantitative and qualitative comparison of our model with best tone mapped outputs over 105 images and also validate our technique through a pair-wise subjective study.

5.2 Related Work

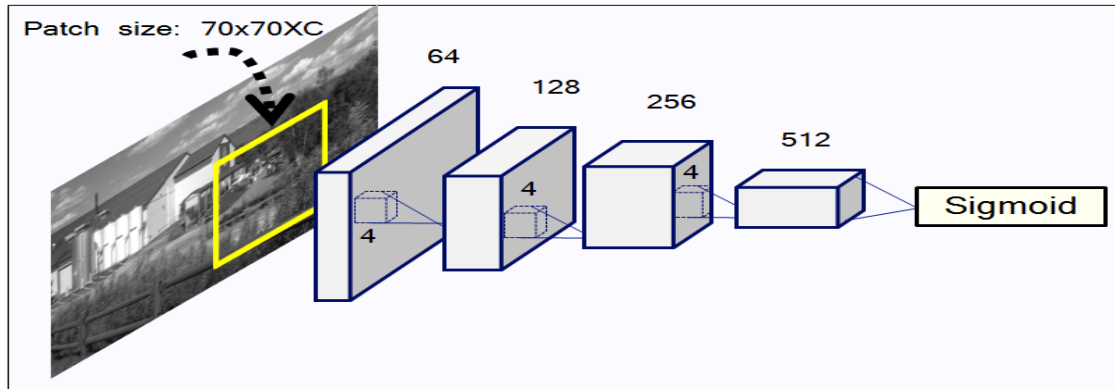
HDR imaging technology has been a subject of interest over the past decades, inspiring to capture and reproduce a wide range of colors and luminous intensities of the real world on a digital canvas. Normally, the information stored in HDR content is represented using a 32-bit floating point format. But to cope with conventional rendering mediums, such scenes are often tone-mapped to an LDR format with available TMOs. A great variety of TMOs addressing different perceptual objectives have been proposed in the past years. In the following, we give a quick review of the tone mapping literature and then would touch upon various deep learning techniques for HDR imaging.

5.2.1 Tone Mapping Operators for HDR Content

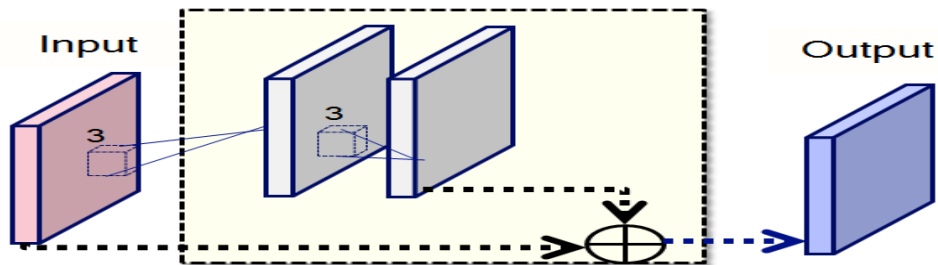
TMOs have been widely explored in the literature, principally based upon how they handle the contrast, color and luminosity in a given HDR image [9]. However, they have been classified into several categories under different sets of criteria [36, 113]. Primarily, they have been grouped into *global* and *local* approaches, relying on how these mapping functions operate on an image. The global methods such as [35, 84, 148] apply the same compression function to all the pixels of an image. For the *local* techniques such as [24, 128, 159], a



(a) Generator (Single Scale).



(b) Discriminator (Single Scale).



(c) Residual Blocks

Figure 5.4 – DeepTMO (single-scale) generator and discriminator architecture. The generator in (a) is an encoder-decoder architecture. Residual blocks in (c) consist of two sequential convolution layers applied to the input, producing a residual correction. Discriminator in (b) consists of a patchGAN [71, 87, 88] architecture which is applied patch wise on the concatenated the input HDR and tone mapped LDR pairs. More details in Appendix section.

tone-mapped pixel depends on the values of its neighboring pixels. Even though global approaches are faster to compute, their resulting LDR outputs do not maintain adequate contrast in the images; thus the scene appears somewhat washed out. The local tone mapping functions, conversely, do not face these issues and are generally capable enough of handling contrast ratios, meanwhile preserving local details. However, these operators result in some prominent ‘halo’ effects around the high frequency edges, thereby giving unnatural artifacts in the scenes. Another category of TMOs [37, 47, 109] includes designs which are inspired from the human visual system, can model the attributes such as adaptation with time, and can discriminate at high contrast stimuli and gradient sensitivities. Nonetheless, all these existing TMOs have been designed to target independently, multiple different objectives [40, 113], such as simulating human visual properties, honest reproduction of scenes, best subjective preference or even for computer vision applications [137]. However, in our work, we mainly focus towards designing a TMO aiming for “best subjective quality output”.

Several small scale perceptual studies have been performed using varied criteria such as with reference or without reference [17, 86, 179] to compare these classical and newly developed TMOs for different perceptual objectives. Even though these subjective studies are ideal to analyze TMO’s performance, the process is bounded to use a limited number of content and TMOs due to practical considerations. As an alternate solution, objective metrics such as [118, 179] have been proposed to automate the evaluation. TMQI is a state-of-the-art objective metric and has been widely used for several TMO optimization studies [29, 100]. It assesses the quality of images on 1) structural fidelity which is a multi scale analysis of the signals, and 2) naturalness, which is derived using the natural image statistics. Both these crucial properties of human perception are combined to define a subjective quality score.

Learning-based methods Parametric sensitivity of hand-crafted TMOs is a well-known phenomenon which impacts the subjective quality of the resulting output. As a result, this emphasizes ‘scene-dependence’ of such tone mapping designs *i.e.* for a given subjective quality task, TMOs have to be fine tuned for each individual scene type. To this end, some optimization based tone mapping frameworks [29, 100] have been designed where the parameters of a specific TMO are optimized for a given image. However, the parameter fine-tuning process for each scene separately is time consuming and limits its real-time applicability. Additionally, it somehow questions the ‘automatic’ nature of tone mapping [40] for their applicability on a wide variety of real-world scenes.

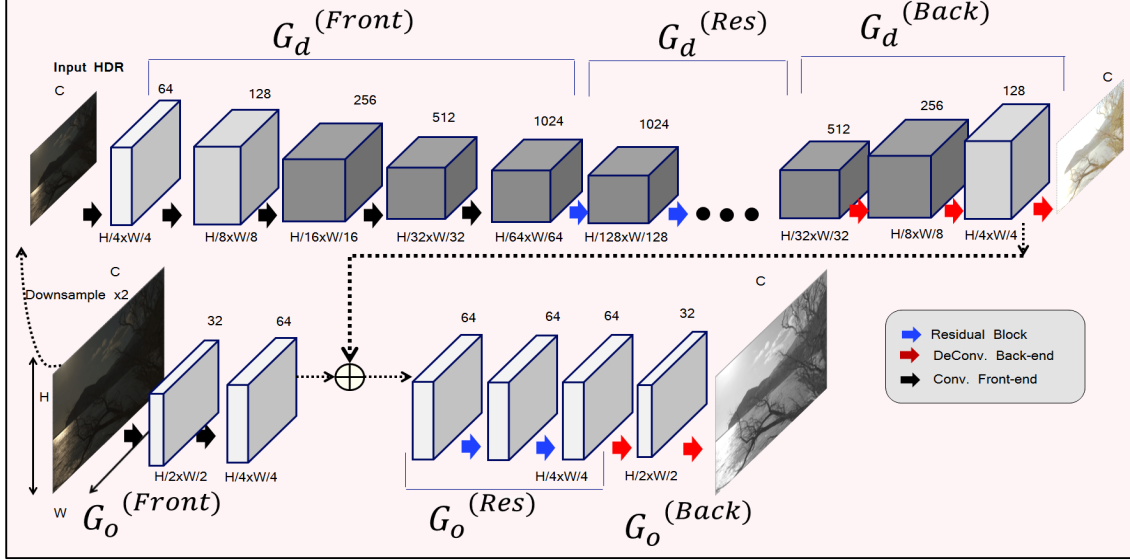


Figure 5.5 – DeepTMO multi-scale generator architecture. While the finer generator G_o has the original image as its input, the input to G_d is a $2\times$ down-sampled version.

5.2.2 CNNs for HDR Scenes

Recently, CNNs have been utilized extensively for multiple HDR imaging tasks such as reconstructing HDR using a single-exposure LDR [39], predicting and merging various high and low exposure images for HDR reconstruction [41] or yielding HDR outputs from dynamic LDR inputs [77]. CNNs have also been modeled to learn an input-output mapping as done for de-mosaicking and de-noising by [52] or learning an efficient bilateral grid for image enhancement [20]. [53] have recently proposed a deep bilateral tone mapper, but it works only for 16-bit linear images and not for conventional 32-bit HDR images. A recent work [65] addresses the end-to-end tone mapping problem where the model is trained for a given scene. This is somewhat similar approach to parameter-tuning where the model is calibrated for only one given scene at a time. Therefore, the problem of designing a fast, parameter-free, end-to-end TMO which can effectively tone map wide variety of real-world high-resolution content for high quality rendering in real time, still holds relevance.

As observed in the past CNN studies, the quality of resulting output depends heavily on the choice of the loss function. Formulating a loss function that constrains the CNN to yield sharp, top quality tone-mapped LDR from their corresponding linear-valued HDR is complex and an ill posed problem. Our work doesn't encounter such an issue as we utilize a GAN based architecture.

5.2.3 Generative Adversarial Networks

GANs [59] have attracted lots of attention owing to their capability of modeling the underlying target distribution by forcing the predicted outputs to be as indistinguishable from the target images as possible. While doing this, it implicitly learns an appropriate loss function, thus eliminating the requirement of hand crafting one by an expert. This property has enabled them to be utilized for wide variety of image processing tasks such as super-resolution [87], photo-realistic style-transfer [75] and semantic image in-painting [180].

For our task, we employ GANs under a conditional setting, commonly referred as cGANs [114], where the generated output is conditioned on the input image. One distinctive feature of cGANs framework is that they learn a structured loss where each output pixel is conditionally dependent on one or more neighboring pixels in the input image. Thus, this effectively constrains the network by penalizing any possible structural difference between input and output. This property is quite useful for our task of tone-mapping where we only want to compress the dynamic range of an HDR image, keeping the structure of the output similar to the input HDR. For this specific reason, cGANs have been quite popular for image-to-image translation tasks, where one representation of a scene is automatically converted into another, given enough training pairs [71] or without them under unsupervised settings [93, 160, 188]. However, a major limitation of using cGANs is that it is quite hard to generate high resolution images due to training instability and optimization issues. The generated images are either blurry or contain noisy artifacts such as shown in Fig. 5.6 (a). In [22], motivated from perceptual loss [75], the authors derive a direct regression loss to generate high-resolution 2048×1024 images, but their method fails to preserve fine-details and textures. In [168], authors have recently shown significant improvement on the quality of high-resolution generated outputs through a multi-scale generator-discriminator design. A similar work for converting HDR to LDR using GANs [126] has also appeared recently where authors oversimplifies the tone-mapping problem by testing only on small 256×256 image crops. Essentially such an approach may not substantially capture the full luminance range present in HDR images thereby overlooks the basic goal of TMO by working on full dynamic range scenes. We however showcase our findings using their adopted architectures from [71] on 1024×2048 HDR images in the Appendix section.

In summary, we motivate the DeepTMO design with these given findings, and discuss the impact of scales for both generator and discriminator, while showcasing their ability to generate high-resolution tone-mapped outputs.

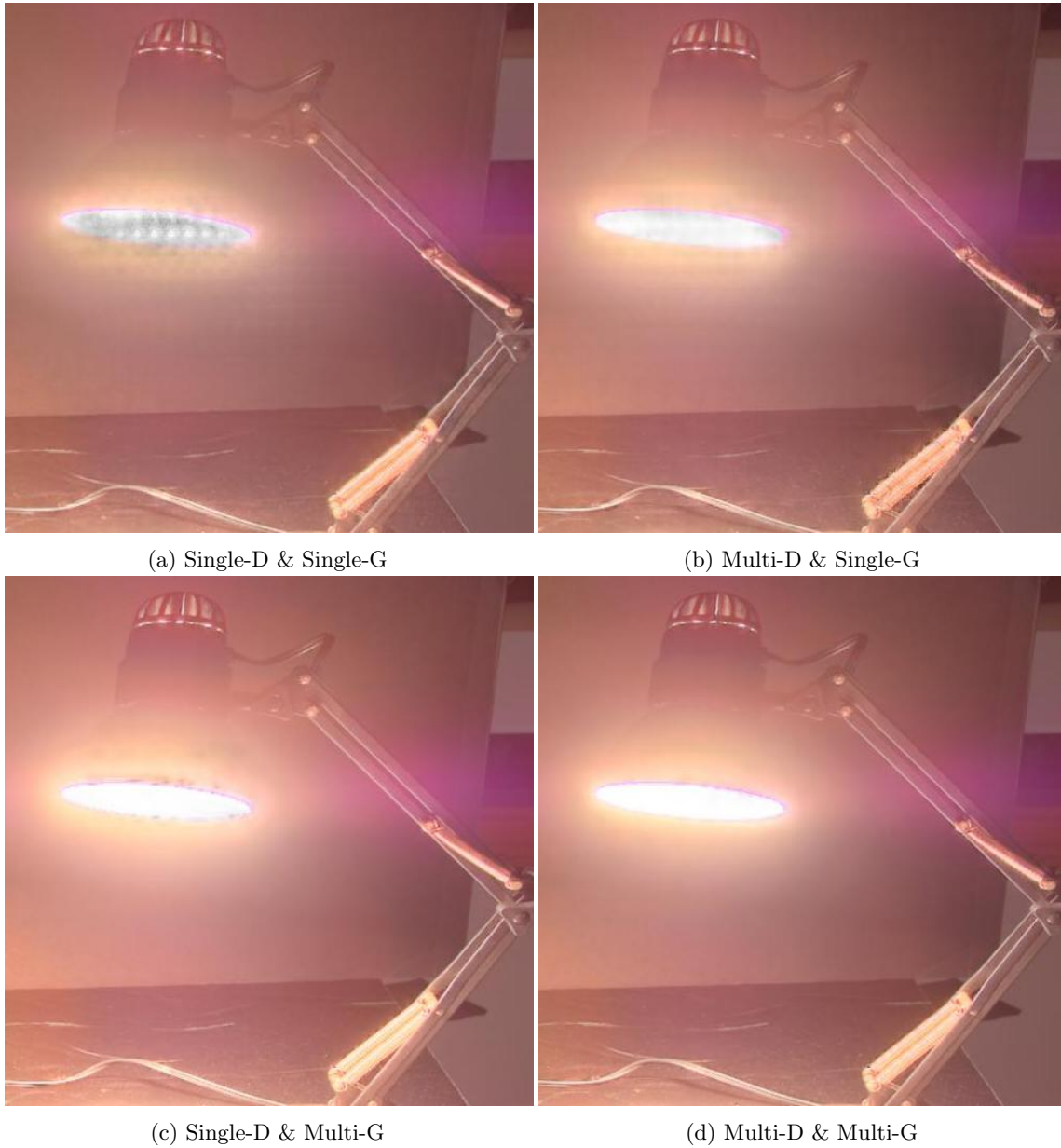


Figure 5.6 – Impact of Multi-scale Discriminator and Generator.

5.3 Algorithm

5.3.1 Problem Formulation

We propose a fast DeepTMO model with the prime objective of producing high-resolution and high-quality tone-mapped images for vast variety of real-world HDR images. Ideally, our model should automatically adapt for each scene without any external parameter tuning. To this end, we propose to imbibe different desired tone mapping characteristics depending upon scene type, content, brightness, contrast etc., to yield high perceptual quality output. In the following paragraphs, we will briefly discuss the formulation of our DeepTMO model.

Linear Domain Input For our models, we directly work on linear values. We performed the scaling to $[0,1]$ with very high-precision, thereby, not impacting the overall output brightness. Additionally, we also experimented with log-scaling the input HDR before performing the tone mapping operation, specifically to test for halo-effects in high-exposure regions such as the sun in Fig.5.15.

Color Reproduction Most classical TMOs perform the dynamic range compression in the luminance channel only and the colors are then reproduced in the post processing stage. This partially accounts to ease the computational complexity of the tone-mapping operation. We follow a similar paradigm, employing the common methodology for color reproduction [148] given as $C_{out} = \frac{C_{in}}{L_{in}} \cdot L_{out}$, where C_{out} and L_{out} are output color and luminance images while C_{in} is the input HDR color image.

Motivation for GANs To achieve the desired TMO, one solution is to use a simple L_1 or perceptual (L_p) loss function [75] with an encoder-decoder architecture as utilized in the past by various inverse-TMOs for generating HDR scenes from single-exposure [39] or multi-exposure [41] LDR images. However, such naive loss functions suffer from either overall spatial blurring (evident in L_1 loss in Fig. 5.1) or over-compression of contrast (evident in L_p loss in Fig. 5.2). This is quite trivial mainly because a CNN architecture learns a mapping from all possible dynamic range values available in the wide variability of training-set scenes to a range $[0,255]$. Thus, the trained model effectively predicts a fairly mean luminance value for most of the pixels in output images to minimize the overall loss function. Another simple idea could be to use TMQI directly as loss function. However, due to the mathematical design of TMQI’s naturalness component and characteristic discontinuity, TMQI cannot be directly used a loss function for back-propagation in a DL framework. In fact, the alternate methodology proposed by authors in [100], which



(a) without VGG and FM Loss



(b) without VGG loss



(c) without FM Loss



(d) with VGG and FM loss

Figure 5.7 – DeepTMO (single-scale) with/without FM and VGG Loss.

optimizes a given TMO using TMQI, is also impossible to be imbibed into an end-to-end DL pipeline, as it treats both SSIM and naturalness separately using two different optimization strategies.

Given the goal of our TMO, designing an effective cost function manually for catering to wide variability of tone-mapping characteristics under different scenic-content is quite a complex task. An alternate solution could be to *learn* such a loss function. The use of GAN is an apt choice here, as it learns an adversarial loss function by itself (the loss being the discriminator network), that encapsulates all the desired features for an ideal TMO encoded in the underlying training data, thereby eradicating the need of manually designing a loss function. Another added advantage of GANs is that they can easily *hallucinate* a richer contrast of pixel luminance values in the tone-mapped outputs compared to reproducing closer to mean luminance values in case of ordinary L_1 / L_p loss functions.

Aiming an artifact-free high-resolution tone-mapped output, we begin investigating the choice of architecture from single-scale to its multi-scale variant for both generator and discriminator in the following sections.



(a) With Instance Norm

(b) With Batch Norm

Figure 5.8 – Batch Normalization vs. Instance Normalization.

5.3.2 DeepTMO (Single-Scale)

Fig. 5.3 depicts an overview of our training algorithm. For our DeepTMO model, we basically employ a cGAN framework [114] which implicitly learns a mapping from an observed HDR image x to a tone mapped LDR image y , given as: $G : x \rightarrow y$. The architecture is composed of two fundamental building blocks namely a discriminator (D)

and a generator (G).

The input to G consists of an $H \times W \times C$ size HDR image normalized between $[0, 1]$. We consider $C = 1$ *i.e.* only the luminance channel is given as an input. Its output is a tone-mapped image (top row of fake pair in Fig. 5.3) of same size as the input. D on the other hand, takes luminance channels of HDR and tone mapped LDR images as input pairs, and predicts whether they are real tone-mapped images or fake. It is trained in a supervised fashion, by employing a training dataset of input HDR and their corresponding *target* tone-mapped images (real-pair in Fig. 5.3). We detail the complete methodology to build our target dataset in Section 5.4. An additional advantage of conditioning on an HDR input is that it empowers D to have some pre-information to make better reasoning for distinguishing between a real or fake tone mapped images, thus accelerating its training.

Next, we discuss the architectures for single-scale generator (Single-G) and single-scale discriminator (Single-D) which are our adaptations from past studies [75, 188] which show impressive results for style transfer and super-resolution tasks on LDR images. Further on, in the subsequent sections, we will reason as to why opting for their multi-scale versions aids in further refining the results.

Generator Architecture (Single-G) The Single-G architecture is an encoder-decoder architecture as shown in Fig. 5.4a. Overall, it consists of a sequence of 3 components: the convolution front end $G^{(Front)}$, a set of residual blocks $G^{(Res)}$ and the deconvolution back end $G^{(Back)}$. $G^{(Front)}$ consists of 4 different convolution layers which perform a subsequent down-sampling operation on their respective inputs. $G^{(Res)}$ is composed of 9 different residual blocks each having 2 convolution layers, while $G^{(Back)}$ consists of 4 convolution layers each of which up-samples its input by a factor of 2. During the down-sampling, $G^{(Front)}$ compresses the input HDR, thus keeping the most relevant information. $G^{(Res)}$ then applies multiple residual corrections to convert the compressed representation of input HDR to one that of its Target LDR counterpart.

Finally, $G^{(Back)}$ yields a full size LDR output from this compressed representation through the up-sampling operation.

Discriminator Architecture (Single-D) The Single-D architecture resembles a 70×70 PatchGAN [71, 87, 88] model, which aims to predict whether each 70×70 overlapping image patch is real or fake, as shown in Fig. 5.4b. Note that the input to D is a concatenation of the HDR and its corresponding LDR image. The Single-D, working on patches, effectively models the high-frequency information by simply restricting its focus upon the structure in local image regions. Moreover, it contains much less parameters compared to a full-image size discriminator, and hence can be easily used for any-size images in a fully convolutional

manner. The Single-D is run across the entire image, and all the responses over various patches are averaged out to yield the final prediction for the image.

Although the Single-G and Single-D architecture yields high-quality reconstructions at a global level, yet it results in noisy artifacts over some specific areas such as bright light sources as shown in Fig. 5.6a. In a way, it necessitates modifying both single-scale versions of G and D to cater not only to coarser information, but at the same time, paying attention to finer level details, thus resulting in a much more refined tone-mapped output.

5.3.3 DeepTMO (Multi-Scale)

While generating high resolution tone-mapped images, it is quite evident now that we need to pay attention towards low-level minute details as well as high-level semantic information. To this end, motivated from [168], we alter the existing DeepTMO (single-scale) model, gradually incorporating step-by-step a multi-scale discriminator (Multi-D) and a multi-scale generator (Multi-G) in the algorithmic pipeline. Different from [168], our adaptation (a) utilizes a 2-scaled discriminator, (b) incorporates a different normalization layer in the beginning given by $\frac{(x-x_{min})}{(x_{max}-x_{min})}$, scaling pixels between $[0,1]$ with high precision, (c) inputs specifically a single luminance channel input with 32-bit pixel-depth linear HDR values.

In the following, we detail the multi-scale versions of G and D . More interestingly, we showcase the impact through step-wise substitution of the Single-D with its Multi-D variant, and then the Single-G as well with its Multi-G counterpart.

Multi-D As shown in Fig. 5.6a and 5.6c, classifying effectively a high-resolution tone-mapped output as real or fake is quite challenging for Single-D. Even though an additional loss term effectively removes noisy artifacts at a global scale in the image (illustrated later in Section 5.3.4), we still witness repetitive patterns in specific localized regions such as around high illumination sources (for *e.g.* inside/outside the ring of table lamp in Fig. 5.6a and on the ring of the lamp in Fig 5.6c). One easy way to tackle this problem is by focusing the discriminator’s attention to a larger receptive field which is possible either through a deeper network or larger convolution kernels. However, it would in turn demand higher memory bandwidth, which is already a constraint while training high resolution HDR images. Thus, we basically retain the same network architecture for the discriminator as used previously, but rather apply it on two different scales of input *i.e.* the original and the $2\times$ down-sampled version, calling the two discriminators D_o and D_d respectively.

Both D_o and D_d are trained together to discriminate between real and synthetically generated images. D_d , by working on a coarser scale, focuses on a larger area of interest in patches throughout the image. This feature subsequently aids G to generate more globally consistent patch-level details in the image. D_o on the other hand, operating at a much finer

scale than D_d , aids in highlighting more precise finer nuances in patches, thus enforcing G to paying attention towards very minute details too at the time of generation. This is elaborated in Fig. 5.6b (where the noisy patterns observed in Single-D & Single-G of Fig. 5.6a, are suppressed to a large extent) and in Fig 5.6d. We still witness minor traces of these artifacts in Fig. 5.6b, due to Single-G’s very own limitations, thus compelling us to switch to Multi-G. Contrary to Single-G, Multi-G reproduces outputs taking notice of both coarser and finer scales. Thus the resultant output having information over both scales, yields a more globally consistent and locally refined artifact-free image as shown in Fig. 5.6b.

Multi-G Fig. 5.5 illustrates the design of Multi-G. It mainly comprises of two sub-architectures, a global down-sampled network G_d and a global original network G_o . The architecture for G_d is similar to Single-G with the components, convolutional front-end, set of residual blocks and convolutional back-end being represented as: $G_d^{(Front)}$, $G_d^{(Res)}$, $G_d^{(Back)}$, respectively. G_o is also similarly composed of three components given by: $G_o^{(Front)}$, $G_o^{(Res)}$ and $G_o^{(Back)}$.

As illustrated in Fig. 5.5, at the time of inference, while the input to G_o is a high resolution HDR image (2048×1024), G_d receives a $2\times$ down sampled version of the same input. G_o effectively makes tone-mapped predictions, paying attention to local fine-grained details (due to its limited receptive field on a high resolution HDR input). At the same time, it also imbibes from G_d , a coarser prediction (as its receptive field has a much broader view). Thus, the final generated output from $G_o^{(Back)}$ encompasses local low-level information and global structured details together in the same tone-mapped output. Hence, what we finally obtain is a much more structurally preserved and minutely refined output which is free from local noisy-artifacts, as seen in Fig. 5.6d.

To summarize, we showcase 4 different cGAN designs where the:

1. Single-D & Single-G architecture encounters noisy patterns due to not paying attention to finer-level details.
2. Multi-D & Single-G architecture is able to suppress to some extent patterns observed in the previous case, though not completely mainly due to limited generalization capabilities of Single-G.
3. Single-D & Multi-G architecture removes patterns throughout the image, however some very localized regions still face artifacts due to the limited capacity of Single-D.
4. Multi-D & Multi-G architecture finally yields us superior quality artifact-free images.

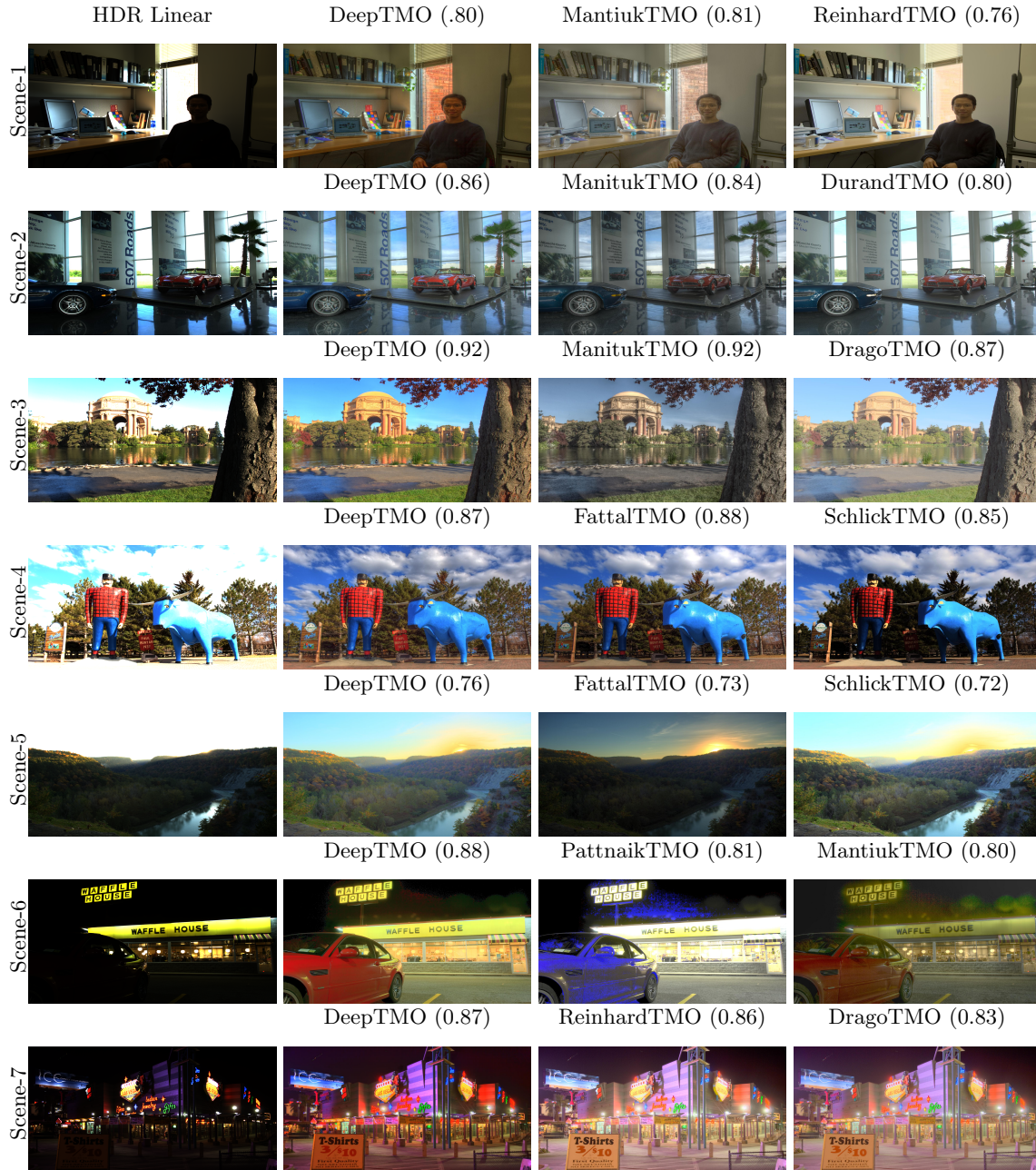


Figure 5.9 – Comparison between our DeepTMO outputs and outputs from top-2 ranked tone-mapped scenes on TMQI metrics for a variety of real-world scenes including indoor, scenes with structures, landscape, dark/noisy scenes. In brackets we show corresponding TMQI scores.

5.3.4 Tone Mapping Objective Function

The ultimate goal of G is to convert high resolution HDR inputs to tone mapped LDR images, while D aims to distinguish real tone-mapped images from the ones synthesized by G . We train both the G and D architectures in a fully supervised setting. For training, we give a set of pairs of corresponding images $\{(x_i, y_i)\}$, where x_i is the luminance channel of the HDR input image while y_i is the luminance output of the corresponding tone-mapped LDR image. Next, we elaborate upon the objective function to train our DeepTMO (both single-scale and multi-scale).

The basic principle behind cGAN [114] is to model the conditional distribution of real tone-mapped images given an input HDR via the following objective:

$$\mathcal{L}_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_x[\log(1 - D(x, G(x)))], \quad (5.1)$$

where G and D compete with each other; G trying to minimize this objective against its adversary D , which tries to maximize it, i.e. $G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D)$.

Since the Multi-D architecture consists of D_o and D_d , our objective for the same is:

$$G^* = \arg \min_G \max_{D_o, D_d} \sum_{s=o,d} \mathcal{L}_{cGAN}(G, D_s) \quad (5.2)$$

We append to the existing cGAN loss, an additional regularization term in the form of a feature matching (FM) loss $\mathcal{L}_{FM}(G, D_s)$ (similar to perceptual loss [34, 51]), given by:

$$\mathcal{L}_{FM}(G, D_s) = \mathcal{E}_{(x,y)} \sum_{i=1}^M \frac{1}{U_i} [||D_s^i(x, y) - D_s^i(x, G(x))||_1], \quad (5.3)$$

where D_s^i is the i^{th} layer feature extractor of D_s (from input to the i^{th} layer of D_s), M is the total number of layers and U_i denotes the number of elements in each layer. In short, we extract features from each individual D layer and match these intermediate representations over real and generated images. Additionally, we append a perceptual loss L_P as used in [75], which constitutes of features computed from each individual layer of a pre-trained 19-layer VGG network [151] given by: $\mathcal{L}_{L_P}(G) = \sum_{i=1}^N \frac{1}{V_i} [||F^{(i)}(y) - F^{(i)}(G(x))||_1]$

where $F^{(i)}$ denotes the i^{th} layer with V_i elements of the VGG network. The VGG network had been pre-trained for large scale image classification task over the Imagenet dataset [144]. Henceforth, our final objective function for a DeepTMO can be written as:

$$\begin{aligned}
G^* = \arg \min_G \max_{D_o, D_d} \sum_{s=o,d} \mathcal{L}_{cGAN}(G, D) + \\
\beta \sum_{s=o,d} \mathcal{L}_{FM}(G, D_s) + \gamma \mathcal{L}_{LP}(G)
\end{aligned} \tag{5.4}$$

β and γ controls the importance of \mathcal{L}_{FM} and \mathcal{L}_{LP} with respect to \mathcal{L}_{cGAN} and both are set to 10 based on emperical validation. We illustrate the impact of both these terms in the following paragraph.

Impact of Feature Matching and Perceptual Loss term Both \mathcal{L}_{FM} and \mathcal{L}_{LP} loss terms act as guidance to the adversarial loss function, thereby preserving overall natural image statistics. As can be seen in Fig 5.7, training without both the terms results in inferior quality throughout the image. After observing Fig. 5.7b and 5.7d, we notice that the VGG-term checks for global noisy repetitive patterns in the image and helps in suppressing them. While being applied on the full generated image, the VGG network captures both low-level image characteristics (*e.g.* fine edges, blobs, colors etc,) and the high level semantic information through its beginning-level and later-stage network layers respectively. Based upon these features, VGG effectively detects the corresponding artifacts as a shortcoming in the overall perceptual quality of the generated scene and hence guides to rectify them, yielding a more natural image. The FM loss term on the other hand, caters to more localized quality details like keeping a watch on illumination conditions in each individual sub-region (*e.g.* it effectively tones down over-exposed regions of windows in the building as deciphered after observing Fig. 5.7c and 5.7d)). This is ideally done by utilizing various feature layers of D , which are trained by focusing upon 70×70 localized image patches. Finally, as shown in Fig. 5.7d, together both VGG and FM loss terms help in yielding a high quality overall contrast and local finer-details preserved output image.

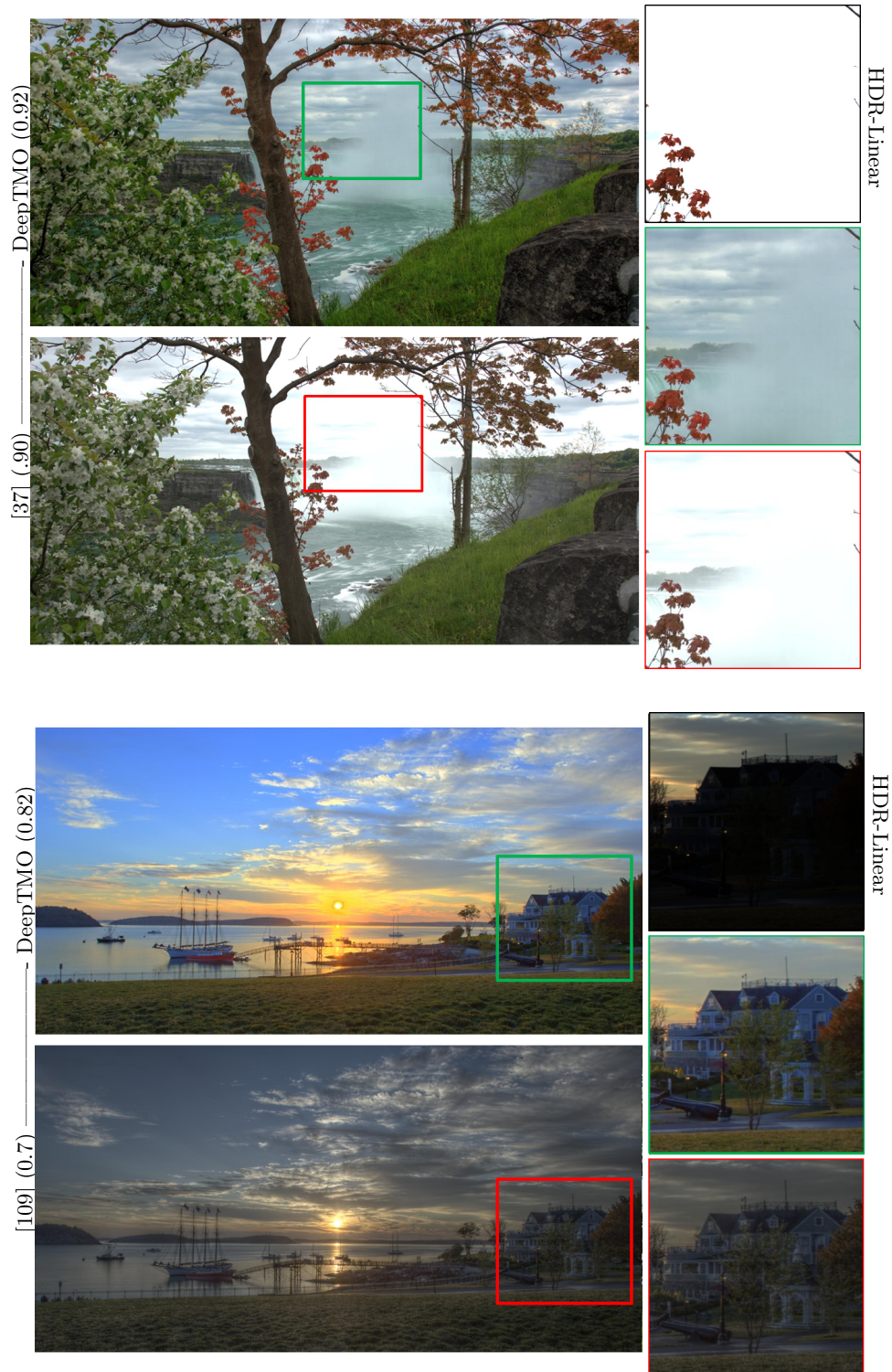


Figure 5.10 – Comparison between DeepTMO and targets, highlighting the zoom-ins with the corresponding HDR-linear input.

5.4 Building the HDR Dataset

In order to design a deep CNN based TMO, it is essential to obtain a large scale dataset with a wide diversity of real-world scenes and cameras. To this end, we gather all the publicly available HDR datasets. For training the network, a total of 698 images are collected from various different sources, listed in the Appendix section. From the HDR video dataset sources, we select the frames manually so that no two chosen HDR images are similar. All these HDR images have been captured from diverse sources which is beneficial for our objective *i.e.* learning a TMO catering a wide variety of real-world scenes.

To further strengthen the training, we applied several data augmentation techniques such as random cropping and flipping, which are discussed briefly in section 5.5. We considered 105 images from the [45] for testing purposes.

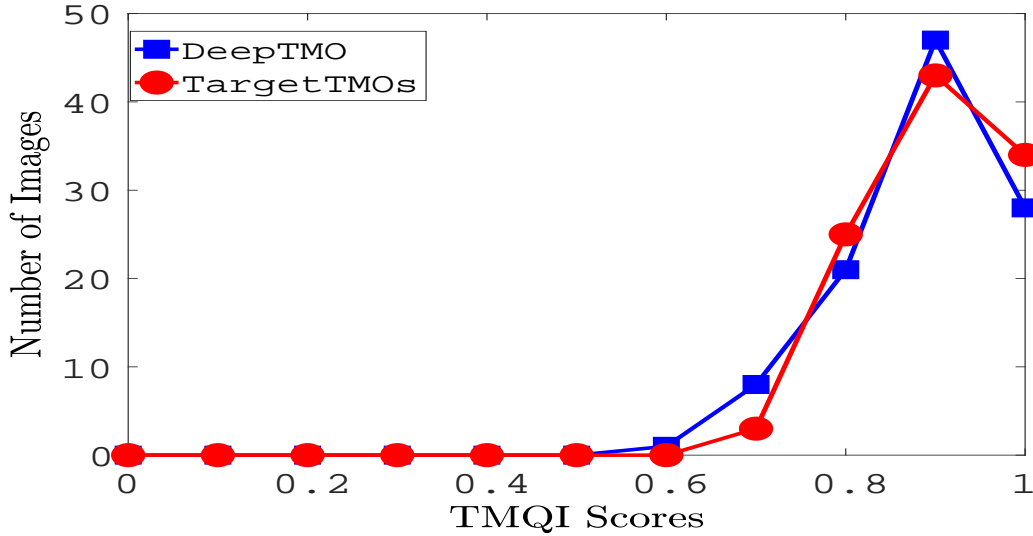


Figure 5.11 – Quantitative performance comparison of best performing DeepTMO with the target TMOs.

5.4.1 Target Tone Mapped Images

Selecting a ‘target’ tone mapped image for an HDR scene is a crucial step in designing the DeepTMO. Although several subjective studies [9] built on different hypotheses have attempted to answer this question, yet they have been conducted only for very small databases of sizes upto 15-20 scenes. Such subjectively evaluated databases are limited in number and cannot be effectively used as training dataset for our DeepTMO model. Additionally, these databases have been evaluated under varying evaluation setups *i.e.* by using different sets of TMOs and reference or no-reference settings. Hence, similar to [126], we resorted to a widely used objective-metric known as TMQI [179] to ensure a fixed target

selection criterion for our problem.

As discussed in Section 2, literature of TMOs is quite extensive and practically difficult to span. Therefore, to find the target tone mapped image for each training HDR scene, we selected 13 classical TMOs: [2, 24, 35, 37, 47, 84, 109, 128, 139, 148, 159] and gamma and log mappings [9]. The selection of these tone mappings is inspired from the subjective evaluation studies [17, 86, 100, 179] which highlight the distinctive characteristics of mapping functions, which we aim to inculcate into the learning of our DeepTMO model.

For each HDR scene, we initially rank the obtained tone-mapped outputs from all the 13 TMOs using the TMQI metric. Then, the best scoring tone mapped output is selected as the ‘target’ for the corresponding HDR scene. Since tuning the parameters of 13 considered TMOs is a daunting task for a large set of training images, we used their default parameter settings throughout this paper. Though we acknowledge that fine-tuning TMO parameters can further boost overall performance, the process however, is almost impractical considering the large amount of training images and the vast parameter-space of the TMOs.

5.5 Training and Implementation Details

The DeepTMO training paradigm is inspired by the conventional GANs approach, where alternate stochastic gradient descent (SGD) steps are taken for D followed by the G . We specifically utilize Least Square GANs (LSGANs), which have proven to yield [111] a much more stable learning process compared to regular GANs. For the multi-scale architecture, we first train G_d separately, and then fine tune both G_d and G_o (after freezing the weights of G_d for the first 20 epochs). For both D and G , all the weights corresponding to convolution layers are initialized using zero mean Gaussian noise with a standard deviation of 0.02, while the biases are set to 0.

Instance Vs. Batch Norm

We use instance normalization [161], which is equivalent to applying batch normalization [70] using a batch size equal to 1. The efficacy of the instance-norm is showcased in Fig. 5.8, where applying the plain batch-norm yields non-uniformity in luminance compression. While the instance normalization is trained to learn mean and standard deviation over a single-scene for the purpose of normalizing, the batch-norm learns over a full batch of input images. Thus, its mean and standard deviation is computed spatially for each pixel from a much wider range of high dynamic luminance values over the entire batch leading to uneven normalization. Absence of batch-norm/instance-norm prevents the G/D to train properly and results in poor generation quality, thus necessitating the need for a normalization layer.

Table 5.1 – *Quantitative Results*. Mean TMQI scores on the test-set of 105 images.

TMOs	TMQI
Tumblin [159] TMO	0.69 \pm 0.06
Chiu [24] TMO	0.70 \pm 0.05
Ashikh [2] TMO	0.70 \pm 0.06
Ward [84] TMO	0.71 \pm 0.07
Log [9] TMO	0.72 \pm 0.09
Gamma [9] TMO	0.76 \pm 0.07
Pattnaik [128] TMO	0.78 \pm 0.04
Schlick [148] TMO	0.79 \pm 0.09
Durand [37] TMO	0.81 \pm 0.10
Fattal [47] TMO	0.81 \pm 0.07
Drago [35] TMO	0.81 \pm 0.06
Reinhard [139] TMO	0.84 \pm 0.07
Mantiuk [109] TMO	0.84 \pm 0.06
DeepTMO (Single G - Single G)	0.79 \pm 0.06
DeepTMO (Single G - Multi D)	0.81 \pm 0.05
DeepTMO (Multi G - Single D)	0.80 \pm 0.07
DeepTMO (Multi G - Multi D)	0.88 \pm0.06

All the instance normalization layers are initialized using Gaussian noise with mean 1 and 0.02 standard deviation.

Implementation

All training experiments are performed using the Pytorch [125] deep learning library with mini-batch SGD, where the batch size is set to 4. For multi-scale, we use batch-size 1 due to limited GPU memory. We utilize an ADAM solver [80] with initial learning rate fixed at 2×10^{-4} for the first 100 epochs and then, allowed to decay to 0.0 linearly, until the final epoch. Momentum term β_1 is fixed at 0.5 for all the epochs. Hyper-parameters have been set to their default values and aren't manipulated much due to GANs training complexity. We also employ random jitters by first resizing the original image to 700×1100 , and then randomly cropping to size 512×512 . For multi-scale, we resize to 1400×2200 and crop to size 1024×1024 . All our networks are trained from scratch.

For all the other handcrafted TMOs, we used the MATLAB-based HDR Toolbox [9] and Luminance HDR software ¹. For each TMO, we enabled the default parametric setting as suggested by the respective authors. Training is done using a 12 Gb NVIDIA Titan-X GPU on a Intel Xeon e7 core i7 machine for 1000 epochs and takes a week.

¹<http://qtpfsgui.sourceforge.net/>

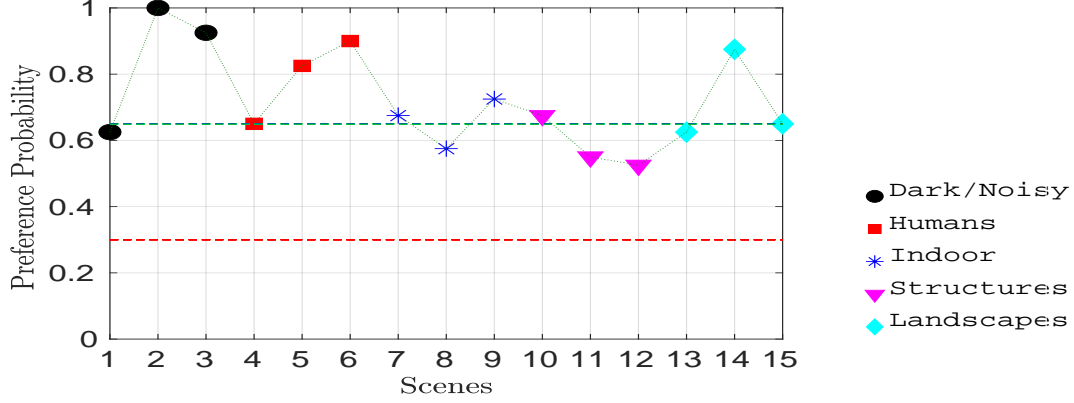


Figure 5.12 – Subjective Test Results. Preference probability of our DeepTMO over best performing target TMOs for 15 scenes representing 5 different scene categories.

5.6 Results and Evaluation

In this section, we present the potential of our DeepTMO on a wide range of HDR scenes, containing both indoor and outdoor, human and structures, as well as day and night views. We compare our results with the best subjective outputs obtained from wide range of tone mapping methods [2, 24, 35, 37, 109, 128, 139, 148] on 105 images of test dataset [45], both qualitatively and quantitatively. In addition, we briefly discuss the specific characteristics of the proposed model, including their adaptation to content or sharpness in rendering high-resolution tone mapped outputs. Finally, we present a subjective evaluation study to access the perceived quality of the output. The size for each input image is kept fixed at 1024×2048 .

Note that test scenes are different from the training set and are not seen by our model while training.

5.6.1 Comparison with the Best Quality Tone-Mapped Images

We begin the comparison of our DeepTMO model against the best quality tone mapped test images to assess the overall capability to reproduce high-quality images over a wide variety of scenes. To obtain the target test image, we follow a similar paradigm as provided in Section 5.4.1.

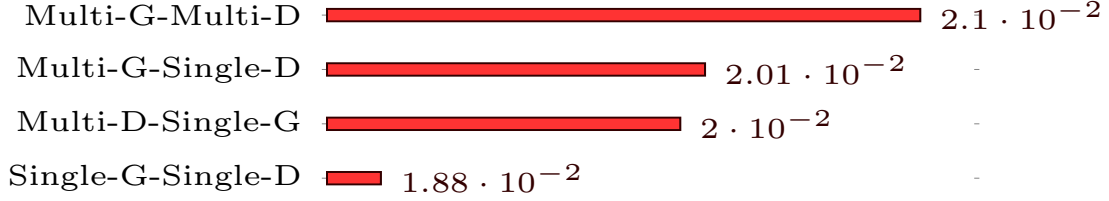
In Fig. 5.9, we demonstrate qualitative comparisons of our model with the two top scoring TMOs obtained using TMQI ranking, which includes methods like Mantiuk [109], Reinhard [139], Fattal [47], Durand [37], Drago [35], Pattnaik [128] TMO, over 7 real-world scenes representing indoor/outdoor, with humans and structures, in day/night conditions. These sample scenes depict the exemplary mapping of linear HDR content using DeepTMO, where it successfully caters a wide variety of scenes as well as competes with the respective

best quality outputs in terms of overall contrast preservation and visual appeal. In scene-1, a scene with human in indoor conditions, we observe that our DeepTMO competes closely to the target output while preserving details in under/over exposed areas such as human face, areas under the table or outside the window. Another indoor scene-2, having shiny surfaces (indoor) and saturated outside regions (windows) demonstrate the effectiveness of our model by preserving details in these regions, yielding a high-quality output. Similar observations can be made in outdoor scenes with structures *i.e.* in scene-3 and 4, where we notice that our DeepTMO model effectively tone-maps sharp frequency regions in overly exposed areas such as the dome of the building, the clouds in the sky or the cow’s body. Landscape scene-5 has similar observations in the rising sun and dark forest regions. Although multi-scale DeepTMO design pays attention to the global and minute sub-regional information, the preservation of illumination and details in dull and overly bright regions is also due to the presence of the FM-loss term, which in turn utilizes features from different D layers. Since D is focused on localized image-patches, the FM-term implicitly understands how to compress or enhance luminance in specific regions.

More interestingly, we observe that DeepTMO suppresses noisy disturbances (*i.e.* above the Waffle House store) in dark scene-6, which appears more pronounced in the two best performing tone-mapped images. This can be reasoned owing to the addition of VGG and FM-loss terms which guides the network to handle the noisy repetitive patterns and dark sensor-noise while preserving the natural scene statistics. Furthermore, we showcase a night time high-contrast scene-7, where our DeepTMO competes closely with the two best quality outputs while preserving the overall contrast ratio. However, we do observe the images obtained with our method have more saturated colors which we discuss later in Section 5.7.1.

Though in most cases our DeepTMO competes well with target images, in some cases we observe that it even outperforms them with respect to TMQI scores. Fig. 5.10 compares two exemplary HDR scenes from the test dataset that are mapped using the DeepTMO and their corresponding target TMOs in day and evening time-settings. In the first row, DeepTMO successfully preserves the fine details in the sky along with the waterfall and the mountains in the background. For a darker evening scene in second row, DeepTMO compensates the lighting and preserves the overall contrast of the generated scene. Even though we observe a halo ring around sun using our method (which we analyze later in Section 5.7.1), our TMQI score is considerably higher mainly because the TMQI metric is color-blind.

Quantitative Analysis To further demonstrate the high-quality mapping capability of DeepTMO models on all the 105 real world scenes, in Fig. 5.11, we show a distribution plot

Figure 5.13 – *Computation time in seconds.*

of the number of scenes against the TMQI Scores. For completeness, we also provide scores achieved by target tone-mapped outputs. The curves clearly show that the generated tone mapped images for DeepTMO compete closely with the best available tone mapped images on the objective metrics with DeepTMO fairing the best amongst all.

We provide quantitative analysis in Table 5.1, to showcase the performance of our proposed model with the existing approaches. For each method, the TMQI scores are averaged over 105 scenes of the test dataset. The final results show that our proposed tone mapping model adapts for the variety of scenes and hence, achieves highest score. Please note that standard TMOs were applied with default parameter settings and hence results may improve for them by parameter optimization. Still performance of our fully automatic approach is highly competitive.

Computation Time Inference is performed on test-images of size 1024×2048 and takes on an average 0.0187 sec. for single-scale and 0.0209 sec. for multi-scale designs, as shown in Figure 5.13.

5.6.2 Quality Evaluation

We performed a subjective pairwise comparison to validate the perceived quality of our tone-mapped images. 20 people participated in this subjective study, with age range of 23-38 years, normal or corrected-to-normal vision.

Test Environment and Setup

The tests were carried out in a room reserved for professional subjective tests with ambient lighting conditions. A Dell UltraSharp 24 Monitor (DELL U2415) was used for displaying images with screen resolution 1920×1200 at 59 hz. The desktop background window was set at 128 gray value.

Each stimuli included a pair of tone mapped images for a given scene, where each pair always consisted of an image produced by DeepTMO and the other one obtained using the best-performing tone mapping functions based on the TMQI rankings. To cater

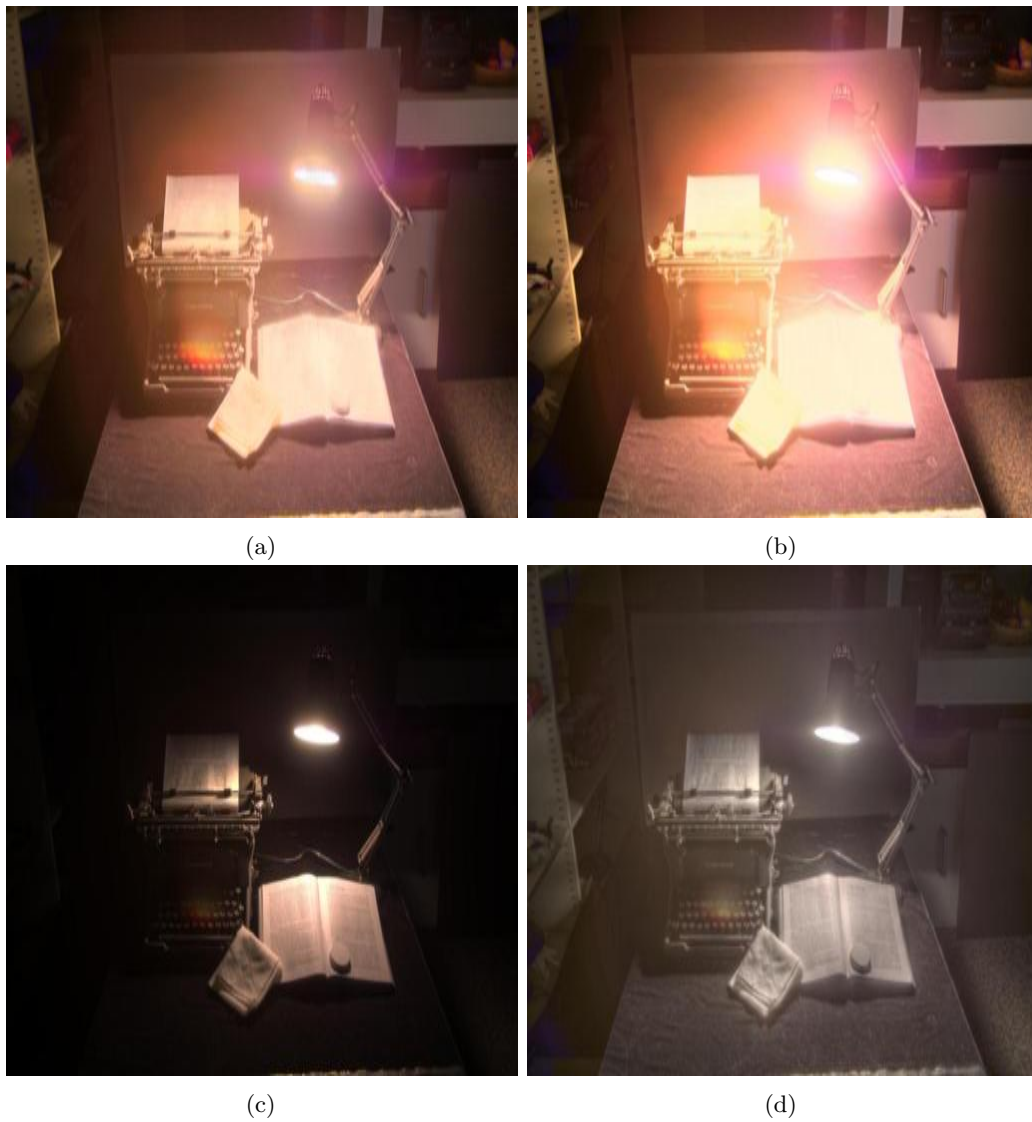


Figure 5.14 – Top TMQI scoring TMOs showing not-so-visually desirable outputs. (a) DeepTMO output, (b), (c) and (d) are 3 top ranking TMO output.

a wide variety of content, we selected 15 scenes from 105 test-set images, representing 5 different categories (3 scenes per category) namely, i) Humans, ii) Dark/Noisy, iii) Indoor, iv) Structures, and v) Landscapes.

Procedures

We conducted a pair-wise subjective experiment where the observer was asked to choose an image by showing a pair of images side-by-side. The option same was not included to force users to choose one of the stimuli. Each participant was asked to select an image which is more realistic and appealing to him/her. Participants were provided with unlimited time to make their decision and record their choice. The experiment was divided into a training and test session, where training involved each participant being briefed to familiarize with the subjective quality evaluation task. Each observer compared a pair of image twice, having each tone-mapped image displayed on both sides (*e.g.* DeepTMO vs. first-best tone mapped and first-best tone mapped vs. DeepTMO).

Results

In order to quantify the results of pairwise subjective tests, we scaled the winning frequencies of the model to the continuous quality-scores using the widely known Bradley-Terry (BT) model in [16]. The scaling is performed using the statistical analysis proposed in [61] to determine whether the perceived visual quality difference of the compared models is statistically significant. The preference probability for our method $Pref - Prob_{(DeepTMO)}$ is mathematically given as:

$$Pref - Prob_{(DeepTMO)} = \frac{w_{DeepTMO}}{N} + \frac{t}{2 \cdot N} \quad (5.5)$$

where $w_{DeepTMO}$ is the winning frequency of our proposed model, t is the tie frequency and N is the total number of participants. The statistical model relies on the hypothesis that each compared TMO in the pairwise test shares equal probability of occurrence *i.e.* 0.5 and hence, follows a Binomial distribution. Based on the initial hypothesis, a Binomial test was performed on the collected data and the critical thresholds were obtained by plotting the cumulative distribution function of the Binomial distribution. By setting 95% as the level of significance, if we receive 13 ($B(13, 20, 0.5) = 0.9423$) or more votes for our proposed method, we consider our tone-mapped image to be significantly favored in terms of subjective quality. Similarly, by setting 5% as the significance level, if we receive 6 ($B(6, 20, 0.5) = 0.0577$) or less votes for our proposed method, we consider our tone-mapped image to be least favored in terms of subjective quality.

The results of the pair-wise subjective quality experiment are shown in Fig. 5.12. The



Figure 5.15 – Halo effect. (a) DeepTMO output, (b) DeepTMO trained with log-scaled values, (c) and (d) 2 top ranking TMO outputs.

two lines (blue and red) mark probabilities of high ($13/20 = .65$) and low ($6/20 = .30$) favor-abilities respectively. Looking at the results, we observe that DeepTMO images have been significantly preferred over best TMQI rated tone mapped images for most of the scenes, for different possible categories. In general, we observed that subjects preferred our tone-mapped LDR scenes which preserve the contrast well. Based on some informal post-experiment interviews, we found that best TMQI rated target images, preserving fine details were least realistic and more like paintings to observers. A small set of images used in subjective tests is shown in Fig. 5.9.

5.7 Conclusion, Limitations and Future work

Designing a fast, automated tone-mapping operator that can reproduce best subjective quality outputs from a wide range of linear-valued HDR scenes is a daunting task. Existing TMOs address some specific characteristics, such as overall contrast ratio, local fine-details or perceptual brightness of the scene. However, the entire process of yielding high-quality tone-mapped output remains a time-consuming and expensive task, as it requires an extensive parameter tuning to produce a desirable output for a given scene.

To this end, we present an end-to-end parameter-free DeepTMO. Tailored in a cGAN framework, our model is trained to output realistically looking tone-mapped images, that duly encompass all the various distinctive properties of the available TMOs. We provide an extensive comparison among various architectural design choices, loss functions and normalization methods, thus highlighting the role that each component plays in the final reproduced outputs. Our DeepTMO successfully overcomes the frequently addressed blurry or tiling effects in recent HDR related works [40, 41], a problem of significant interest for several high-resolution learning-based graphical rendering applications as highlighted in [40]. By simply learning an HDR-to-LDR cost function under a multi-scale GANs framework, DeepTMO successfully preserves desired output characteristics such as underlying contrast, lighting and minute details present in the input HDR at the finest scale. Lastly, we validate the versatility of our methodology through detailed quantitative and qualitative comparisons with existing TMOs.

5.7.1 Limitations and Future Work

Target Selection Though DeepTMO successfully demonstrates versatility in addressing wide variety of scenes, its expressive power is limited by the amount of available training data and quality of its corresponding ‘target’. As noted in Section 5.1, due to unavailability of subjectively annotated ‘best tone mapped images’ for HDR scenes, we resort to an objective TMQI metric to build the corresponding target LDR. However, the metric itself

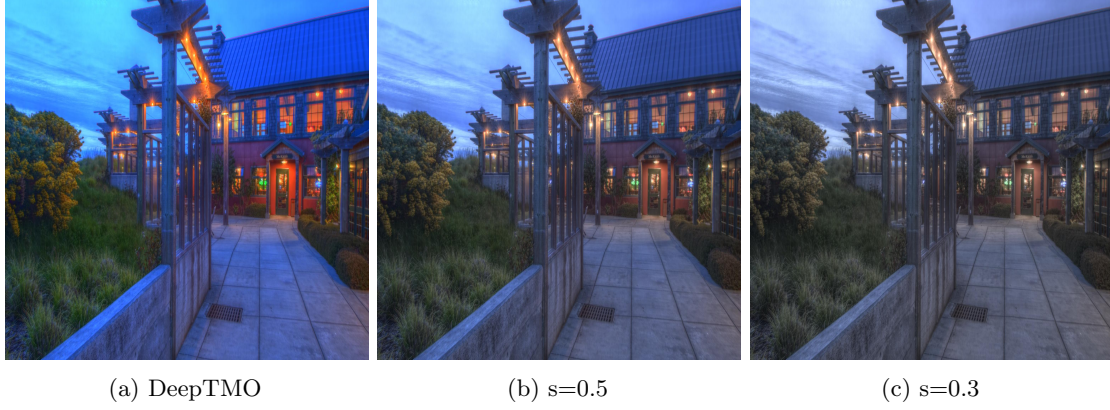


Figure 5.16 – Color Correction. (a) DeepTMO, (b) and (c) are the color corrected DeepTMO controlled by parameter s from [108].

is not as perfect as the human visual system. We illustrate this point in Fig. 5.14. The images ranked lower by TMQI metric in column 3 and 4 are somehow more interesting than their best-ranked counterpart in column 2. Such samples can eventually restrict the generation power of our model.

Another specific case includes ‘Halo’ artifacts or rings around high illumination regions such as the sun shown in Fig. 5.15, where DeepTMO (column 1) is compared with the top TMQI scoring outputs in column 3 and column 4. This is mainly due to the inadequate amount of training data consisting of such samples, and the presence of their overly saturated ‘target’ counterparts. As a result, D has very little information about effectively tone-mapping such regions, and thus is unable to guide G to effectively eradicate such effects at generation time. To handle such artifacts, we additionally experimented using a log-scale input (column 2) where we observe that even log-scale values do not rectify such effects, thus necessitating the need of adequate training samples.

Color Correction Color is an important aspect while rendering high quality subjective tone-mapped outputs. Our proposed method has been trained for efficient luminance compression in HDR scenes and uses the classical color ratios to produce the resulting tone-mapped outputs. Although it provides best subjective quality outputs in most cases, it sometimes can result into overly saturated colors which might look unnatural and perceptually unpleasant. One simple solution could be to simply plug-in existing color correction methods [108] to obtain the desired output. An example is shown in Fig. 5.16, where color correction has been carried out using the method as proposed in [108], which is given by $C_{out} = ((\frac{C_{in}}{L_{in}} - 1) \cdot s + 1) \cdot L_{out}$, where s is the color saturation control. Alternately, another interesting solution could be to learn a model to directly map the content from HDR color-space to an LDR color tone mapped output.

5.8 Appendix

5.8.1 DeepTMO (Single-Scale) Architecture

In this section we specify the detailed architectural details of basic single-scale generator and discriminator.

Generator Architecture

$G^{(Front)}$ has first a convolution layer consisting of 64 filter kernels (or output channels) each of size 7×7 applied with a stride of (1,1) and padding (0,0). Next, there are four convolution layers with 128, 256, 512 and 1024 filter kernels respectively each with a size 3×3 and stride (2,2) and padding (1,1). Each of these four layers are followed by the batch norm with batch size = 1 (also called instance normalization [161]) and Relu [119]. Following this, we have $G^{(Res)}$ which is a set of 9 residual blocks, each of which contains two 3×3 convolutional layers, both with 1024 filter kernels. Next, for $G^{(Back)}$ there are four de-convolutional or transposed convolution layers with 512, 256, 128, 64 filter kernels, each having a filter size of 3×3 and fraction strides of $\frac{1}{2}$. Both these layers have instance normalization and Relu added after the convolution. Finally, there is another convolution layer of size 7×7 and stride 1 followed by a tanh activation function at the end.

Discriminator Architecture

Discriminator architecture consists of 4 convolution layers of sizes 4×4 and stride (2,2). From first to the last, the number of filter kernels is 64, 128, 256 and 512 respectively. Each of the convolutional layer is appended with an instance normalization (except the first layer) and then leaky ReLU [102] activation function (with slope 0.2). Finally, a convolutional layer is applied at the end to yield a 1 dimensional output which is followed by a sigmoid function.

5.8.2 DeepTMO-R With/Without Skip Connections

We additionally explored the cGAN based network from [71] for our tone-mapping task and name it as DeepTMO-R. The generator architecture of DeepTMO-R is shown in 5.17. We use the same discriminator as of DeepTMO single-scale. DeepTMO-R design is altered by adding skip-connections between each layer i and layer $n - i$, n being the total number of encoder-decoder layers and called as DeepTMO-S, which as a result concatenates all the channels at layer i with layer $n - i$. Various past HDR reconstruction methods, have used skip connections [142] for generating HDR scenes from single exposure [40] or multi-exposure [41] LDR images. The basic idea had been that since, both LDR and HDR

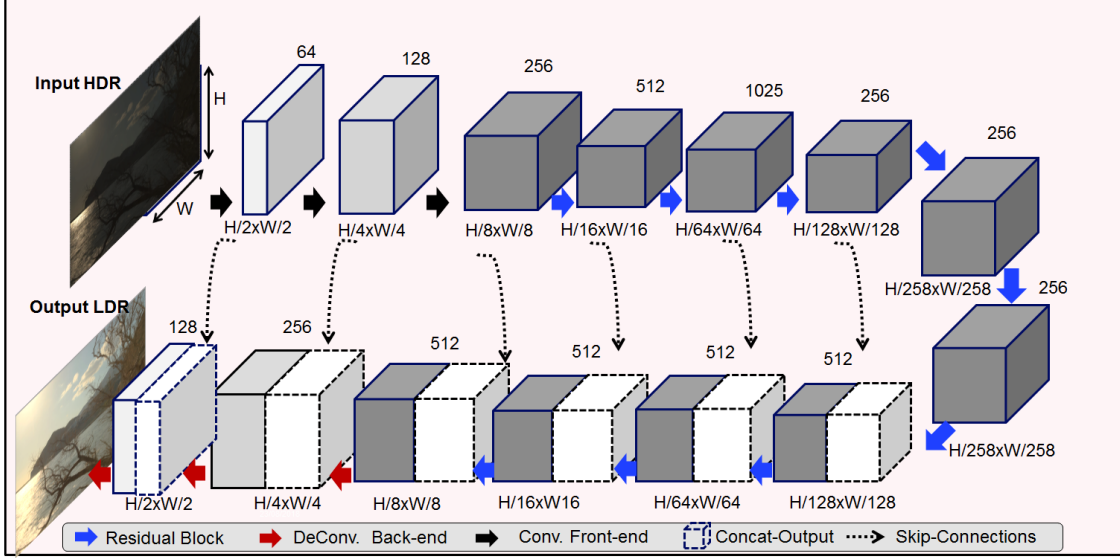


Figure 5.17 – DeepTMO-R and DeepTMO-S generator architecture.

scenes are different renderings of the same underlying structure, at a particular scale, their structures are also more or less aligned. Hence, it is possible to effectively transmit low-level details from input to output scenes, circumventing the bottleneck of the encoder-decoder architecture.

Experimentation

We performed the training and testing for both, DeepTMO-R and DeepTMO-S in a similar fashion to that of DeepTMO and also on the high-resolution images of size 1024×2048 . In Fig. 5.18, we show a simple case of a daytime natural scene where the three architectures provide results with some prominent visible effects. From the cropped insets, we see that DeepTMO-R results in blurry effect on the textured bark of the tree, similar to previous example. DeepTMO-S on the other hand, doesn't produce any blurriness, but instead, we notice pronounced repetitive checkerboard artifacts. Such artifacts have been recently discussed in deep-learning based image rendering problems [50, 120] and are mainly caused due to no direct relationship among intermediate feature maps generated in de-convolutional layers. Nevertheless, it is still an open problem. DeepTMO, on the other hand, gives us sharper and checkerboard free images while preserving the fine-details too.

5.8.3 DeepTMO (Multi-Scale) Architecture

Multi-Scale Generator Architecture

$G_1^{(F)}$ here consists of 5 convolution layers, with the number of output channels from first till last being 64, 128, 256, 512, 1024 respectively. $G_1^{(R)}$ has 9 residual blocks each having 1024



Figure 5.18 – While the DeepTMO-R simply results in blurred outputs in the bark of tree, the DeepTMO-S tries to refine them but is faced by *checkerboard* artifacts [50, 120]. The DeepTMO provides best results amongst the three methods while preserving the fine details, contrast and sharpness in the image.

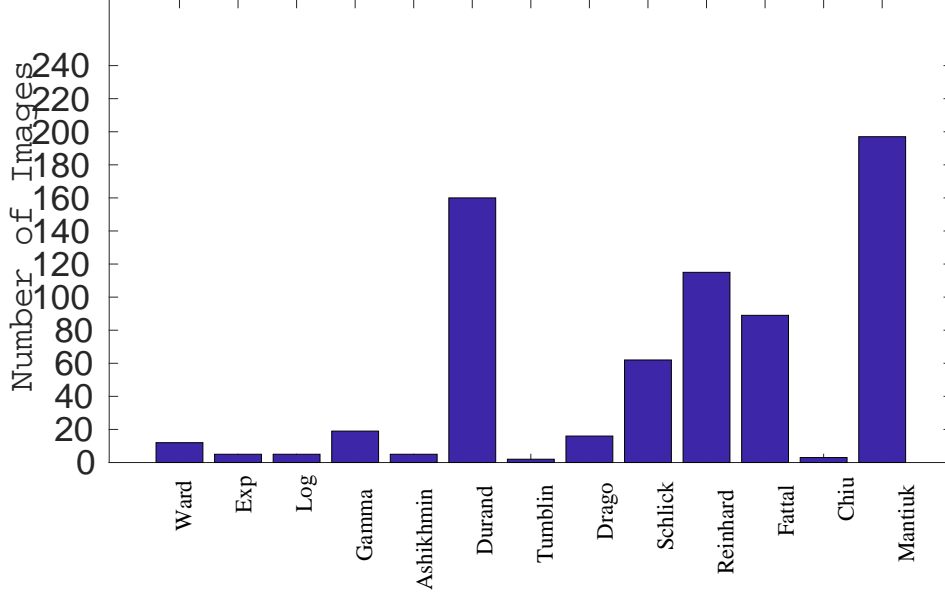


Figure 5.19 – Distribution of best tone mapped output on training dataset of 700 Images.

as the number of output channel. For $G_1^{(B)}$, we have 5 transposed convolution layers with output channels 512, 256, 128, 64 and 1. All the component layers have similar nomenclature as used in the component layers of G , including the first layer of $G_1^{(F)}$ and the last layer in $G_1^{(B)}$.

$G_2^{(F)}$ has two consecutive convolution layers, with output channels 32 and 64 respectively. The last feature map of $G_1^{(B)}$ (with 64 output channels) is then element-wise summed with the output feature map of $G_2^{(F)}$, to provide corresponding input to the residual block $G_2^{(R)}$. $G_2^{(R)}$ consists of 3 residual blocks each with 64 output channels. Following this we have two deconvolution layer in $G_2^{(B)}$ with 32 and 3 output channels respectively. Again the structure of all the component-wise layers is similar to G .

5.8.4 Training dataset

We provide the training distribution of target tone-mapped images on training dataset in Fig. 5.19 considering 13 TMOs. Note that while training we have used several data augmentation techniques. These target tone-mapped scenes have been selected using the default parametric settings.

5.8.5 Dataset Source

Dataset is collected from the following sources: [1, 7, 28, 30, 44, 49, 81, 130, 133, 136, 174]

Chapter 6

Conclusion and future work

6.1 Summary

This thesis presents various applications of deep learning techniques for processing high-resolution RGB images. The application scenarios are broadly categorized into a) Remote-Sensing imagery and b) High Dynamic Range Imaging. Underneath, we highlight the four major contributions of this thesis:

- We first address the problem of scene complexity in Aerial scene Classification. We argue that complex aerial scenes consist of large number of distinctive objects, many of which cause ambiguity in the overall decision making of the neural network. Thus we adopt a deep weakly supervised learning technique to automatically select, simply by using the image-level labels, the most relevant regions in the aerial scene which helps in improving the overall performance of the network. We showcase both quantitatively and qualitatively, how this methodology helps in increasing the classification accuracy for more complex scenes, thus proving our hypothesis.
 - Next, we deal with another challenging yet highly relevant problem of building-footprint extraction in aerial images. Most of the past works dealing with building segmentation rely on refining the segmentation masks either using conditional-random-fields using hand-crafted priors or using other sources such as open-street-maps. We instead propose to refine the initial segmentation prediction by relying on the joint distribution of input and output variables. We effectively do this, by predicting the error probability maps of initial segmentation mask given the input image. Further on, our model replaces only those pixels in the segmentations which have a high error probability score in the previously generated error probability maps. Not only do we see quantitative improvement compared to benchmark results, the qualitative results also assert that the model is better able to learn the underlying structures of building
-

footprints and hence is able to effectively refine them.

- Subsequently, we deal with the problem of removal of thin clouds in satellite images. We adopt a novel-technique utilizing GANs to map the cloudy-images to their non-cloudy counterpart particularly for Sentinel-2 RGB images over the region of Paris. Since, we lack a proper ground-truth of cloud-free images for the cloudy counter-parts, we tackle the issue by learning the mapping in an unsupervised fashion. Apart from the adversarial loss, we additionally utilize a cyclic loss, that constrains the model to generate cloud-free image which respects the input image and not reproduce any random image from the target domain. We compare our results qualitatively with other methods that use single image haze removal where we show huge gain in producing cloud-free images. Due to lack of ground-truth, we compare quantitatively on synthetically generated images, where our method yields much higher PSNR and SSIM scores compared to other methods.
- Lastly, we tackle a slightly different problem settings, dealing with tone mapping of High Dynamic Range Imaging in order to yield high resolution and high subjective quality LDR images over a wide variety of scenes. We do this, by adopting 4 different conditional GAN settings to model the tone mapping function and conclude that a multi-scale generator discriminator network design is the best suitable for reproducing high resolution and high subjective quality tone mapped outputs. Due to lack of ground truth tone mapped dataset, we generate the targets using the objective TMQI metrics. We finally showcase quantitative as well as subjective tests compared to other tone mapping operators, thus proving the superiority of our tone-mapped outputs.

6.2 Future Research Directions

We propose several future research directions as an extension of our past work. We discuss some of them below.

Aerial scene segmentation using weakly supervised learning Generating ground truth pixel-based labels for aerial scenes is highly challenging and tedious task. Moreover it requires expert knowledge and often suffers from insufficient captured information depending upon the sensor. Under such a setting, it is quite relevant to perform such labeling, either in an unsupervised or weakly supervised fashion by using only image-level labels. This would also be relevant for correction of Open street maps where we can easily notice registration errors between the segmentation mask and the underlying RGB imagery.

Cloud Removal for effective Land-Cover Classification Classifying Land-cover is an important task for estimating green-cover or amount of urbanizing or water sources in an region. However they are often plagued by clouds that inhibit in clearly visualizing the underlying region. The work presented in this thesis can effectively help in removing thin cloud-cover in an unsupervised fashion using the Cloud-GAN model that we discussed in Chapter-3. For thick cloud-cover we can condition the GAN network on an additional cloud-penetrating source, thus giving us rich information about the underlying region.

Tone mapping of HDR scenes for color images Tone mapping performed using our DeepTMO has two major issues in the current training pipeline. The first is that the TMQI metrics which has been used to rank tone mapping operators is color blind. Hence a TMO which has a better color image might be ranked below one which has an inferior color output. Thus an interesting future research direction can be to design a metrics which can rank the color images instead of luminance scale. Second issue is that, our model at times, yields saturated color outputs as well as halo artifacts, as it has been trained for luminance channel and not for all RGB channels. Thus another future work can be to design a model which can predict color outputs addressing these two issues.

Publications

Journal articles

1. *P. Singh, N. Komodakis, Improving recognition of complex aerial scenes using a deep weakly supervised learning paradigm (IEEE Geoscience and Remote Sensing Letters, 2018)*
2. *P. Singh, N. Komodakis, Refining segmentations of buildings and roads through a novel deep structured prediction methodology (to be submitted to Computer Vision and Image understanding Journal)*
3. *P. Singh, N. Komodakis, Removing thin cloud cover in Sentinel-2 imagery using Cyclic Generative Adversarial Networks. (submitted to IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing)*
4. *P.Singh*, A.Rana*, G. Valenzise, F. Dufaux, N. Komodakis, A. Smolic: Deep Tone Mapping for High Dynamic Range Scenes (submitted to IEEE Transactions on Image Processing)*

Conference papers

1. *P. Singh, N. Komodakis, Effective Building Extraction by Learning to Detect and Correct Erroneous Labels in Segmentation Masks, IGARSS 2018 (Oral Presentation)*
 2. *P. Singh, N. Komodakis, Cloud-GAN: Cloud Removal For Sentinel-2 Imagery using a Cyclic Consistent Generative Adversarial Networks, IGARSS 2018 (Oral Presentation)*
-

References

- [1] W. J. Adams, J. H. Elder, E. W. Graf, J. Leyland, A. J. Lugtigheid, and A. Murry, “The southampton-york natural scenes (syms) dataset: Statistics of surface attitude,” 2016. *Cited in Sec.* 5.8.5
 - [2] M. Ashikhmin, “A tone mapping algorithm for high contrast images,” pp. 145–156, 2002. *Cited in Sec.* 5.4.1, 5.1, 5.6
 - [3] N. Audebert, B. L. Saux, and S. Lefèvre, “How useful is region-based classification of remote sensing images in a deep learning framework?” in *IGARSS*, July 2016, pp. 5091–5094. *Cited in Sec.* 4.1
 - [4] —, “Fusion of heterogeneous data in convolutional networks for urban semantic labeling,” in *2017 Joint Urban Remote Sensing Event (JURSE)*, March 2017, pp. 1–4. *Cited in Sec.* 3.1
 - [5] N. Audebert, B. Le Saux, and S. Lefèvre, “Joint Learning from Earth Observation and OpenStreetMap Data to Get Faster Better Semantic Maps,” in *EARTHVISION 2017 IEEE/ISPRS CVPR Workshop. Large Scale Computer Vision for Remote Sensing Imagery*, Honolulu, United States, Jul. 2017. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01523573> *Cited in Sec.* 3.1, 3.2
 - [6] —, “Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks,” *ISPRS Journal of Photogrammetry and Remote Sensing*, 2018. *Cited in Sec.* 2.1
 - [7] M. Azimi, A. Banitalebi-Dehkordi, Y. Dong, M. T. Pourazad, and P. Nasiopoulos, “Evaluating the performance of existing full-reference quality metrics on high dynamic range (hdr) video content,” in *International Conference on Multimedia Signal Processing (ICMSP)*, 2014. *Cited in Sec.* 5.8.5
 - [8] Y. Bandoh, G. Qiu, M. Okuda, S. Daly, T. Aach, and O. C. Au, “Recent advances in high dynamic range imaging technology,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 3125–3128. *Cited in Sec.* 1
 - [9] F. Banterle, A. Artusi, K. Debattista, and A. Chalmers, *Advanced High Dynamic Range Imaging: Theory and Practice*, February 2011. *Cited in Sec.* 5.2.1, 5.4.1, 5.1, 5.5
 - [10] N. Barakat, A. N. Hone, and T. E. Darcie, “Minimal-bracketing sets for high-dynamic-range image capture.” *IEEE Trans. Image Processing*, vol. 17, no. 10, pp. 1864–1875, 2008. *Cited in Sec.* 1
 - [11] L. Bashmal, Y. Bazi, H. AlHichri, M. M. AlRahhal, N. Ammour, and N. Alajlan, “Siamese-gan: Learning invariant representations for aerial vehicle image categorization,” *Remote Sensing*, vol. 10, no. 2, p. 351, 2018. *Cited in Sec.* 4.2.3
 - [12] D. Berman, T. Treibitz, and S. Avidan, “Non-local image dehazing,” in *IEEE Conf. CVPR*, 2016. *Cited in Sec.* 4.2, 4.3, ??, 4.8i, 4.6.2, 4.6.3
 - [13] —, “Air-light estimation using haze-lines,” in *IEEE Conf. ICCP*, 2017. *Cited in Sec.* 4.2, 4.3, ??, 4.8i, 4.6.2, 4.6.3
 - [14] X. Bian, C. Chen, L. Tian, and Q. Du, “Fusing local and global features for high-resolution scene classification,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2889–2901, June 2017. *Cited in Sec.* ??
 - [15] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, “Multi-task learning for segmentation of building footprints with deep neural networks,” *arXiv preprint arXiv:1709.05932*, 2017. *Cited in Sec.* 3.1, 3.2, ??, 3.5.1, 1
-

- [16] R. A. Bradley and M. E. Terry, "The rank analysis of incomplete block designs — I. The method of paired comparisons," *Biometrika*, vol. 39, pp. 324–345, 1952. *Cited in Sec. 5.6.2*
- [17] M. Cadik, M. Wimmer, L. Neumann, and A. Artusi, "Evaluation of HDR tone mapping methods using essential perceptual attributes," *Computers and Graphics*, pp. 330 – 349, 2008. *Cited in Sec. 5.2.1, 5.4.1*
- [18] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *arXiv preprint arXiv:1508.00092*, 2015. *Cited in Sec. 1*
- [19] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for vhr remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4775–4784, Aug 2017. *Cited in Sec. ??*
- [20] J. Chen, A. Adams, N. Wadhwa, and S. W. Hasinoff, "Bilateral guided upsampling," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 203, 2016. *Cited in Sec. 5.2.2*
- [21] J. Chen, X. Zhu, J. E. Vogelmann, F. Gao, and S. Jin, "A simple and effective method for filling gaps in landsat etm+ slc-off images," *Remote sensing of environment*, vol. 115, no. 4, pp. 1053–1064, 2011. *Cited in Sec. 4.2.1*
- [22] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017. *Cited in Sec. 5.2.3*
- [23] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 415–423. *Cited in Sec. 1*
- [24] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman, "Spatially nonuniform scaling functions for high contrast images," in *Proceedings of Graphics Interface '93*, ser. GI '93, Toronto, Ontario, Canada, 1993, pp. 245–253. *Cited in Sec. 5.2.1, 5.4.1, 5.1, 5.6*
- [25] P. R. Coppin and M. E. Bauer, "Digital change detection in forest ecosystems with remote sensing imagery," *Remote sensing reviews*, vol. 13, no. 3-4, pp. 207–234, 1996. *Cited in Sec. 1*
- [26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223. *Cited in Sec. 1*
- [27] D. G. CosmiQWorks and NVIDIA, "Spacenet," <http://explore.digitalglobe.com/spacenet>, 2016. *Cited in Sec. 3.1*
- [28] M. Database. (2004) Mpi hdr image database. [Online]. Available: <http://resources.mpi-inf.mpg.de/hdr/gallery.html> *Cited in Sec. 5.8.5*
- [29] K. Debattista, "Application specific tone mapping via genetic programming," *Computer Graphics Forum*, vol. 37, no. 1, pp. 439–450, 2017. *Cited in Sec. 5.1, 5.2.1*
- [30] P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997. *Cited in Sec. 5.8.5*
- [31] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, pp. 31–71, Jan. 1997. *Cited in Sec. 2.1, 2.2.3*
- [32] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199. *Cited in Sec. 1*
- [33] W. Dong, X. Zhang, and C. Zhang, "Generation of cloud image based on perlin noise," in *2010 International Conference on Multimedia Communications*, Aug 2010, pp. 61–63. *Cited in Sec. 4.5.1*

- [34] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Advances in Neural Information Processing Systems*, 2016, pp. 658–666. *Cited in Sec.* 5.3.4
- [35] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, “Adaptive logarithmic mapping for displaying high contrast scenes,” *Computer Graphics Forum*, pp. 419–426, 2003. *Cited in Sec.* 5.2.1, 5.4.1, 5.1, 5.6, 5.6.1
- [36] F. Dufaux, P. Le Callet, R. Mantiuk, and M. Mrak, *High Dynamic Range Video: From Acquisition, to Display and Applications*. Academic Press, 2016. *Cited in Sec.* 5.1, 5.2.1
- [37] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” in *29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’02, 2002. *Cited in Sec.* 5.2.1, ??, 5.4.1, 5.1, 5.6, 5.6.1
- [38] T. Durand, N. Thome, and M. Cord, “WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks,” in *CVPR*, 2016. *Cited in Sec.* 2.2.2, 2.2.3
- [39] G. Eilertsen, J. Kronander, G. Denes, R. K. Mantiuk, and J. Unger, “Hdr image reconstruction from a single exposure using deep cnns,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, p. 178, 2017. *Cited in Sec.* 1, 5.2.2, 5.3.1
- [40] G. Eilertsen, R. Wanat, R. K. Mantiuk, and J. Unger, “Evaluation of Tone Mapping Operators for HDR-Video,” *Computer Graphics Forum*, 2013. *Cited in Sec.* 5.1, 5.2.1, 5.7, 5.8.2
- [41] Y. Endo, Y. Kanamori, and J. Mitani, “Deep reverse tone mapping,” *ACM Transactions on Graphics (Proc. of SIGGRAPH ASIA 2017)*, vol. 36, no. 6, nov 2017. *Cited in Sec.* 1, 5.1, 5.2.2, 5.3.1, 5.7, 5.8.2
- [42] K. Enomoto, K. Sakurada, W. Wang, H. Fukui, M. Matsuoka, R. Nakamura, and N. Kawaguchi, “Filmy cloud removal on satellite imagery with multispectral conditional generative adversarial nets,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017, pp. 1533–1541. *Cited in Sec.* 4.1, 4.2.3, 4.5.1, 4.6.2
- [43] E. S. A. (ESA), “Sentinel,” <https://sentinel.esa.int/web/sentinel/home>, 2014. *Cited in Sec.* 1, 3.1
- [44] ETHyma. (2015) Ethyma database for high dynamic range images. [Online]. Available: <http://ivc.univ-nantes.fr/en/databases/ETHyma/> *Cited in Sec.* 5.8.5
- [45] M. Fairchild. (2007) The hdr photographic survey. [Online]. Available: <http://www.rit-mcsl.org/fairchild/HDR.html> *Cited in Sec.* 5.4, 5.6
- [46] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning hierarchical features for scene labeling,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013. *Cited in Sec.* 1
- [47] R. Fattal, D. Lischinski, and M. Werman, “Gradient domain high dynamic range compression,” *ACM Trans. Graph.*, vol. 21, no. 3, pp. 249–256, Jul. 2002. *Cited in Sec.* 5.2.1, 5.4.1, 5.1, 5.6.1
- [48] G. Ferrier, “Application of imaging spectrometer data in identifying environmental pollution caused by mining at rodaquilar, spain,” *Remote Sensing of Environment*, vol. 68, no. 2, pp. 125–137, 1999. *Cited in Sec.* 1
- [49] J. Froehlich, S. Grandinetti, B. Eberhardt, S. Walter, A. Schilling, and H. Brendel, “Creating cinematic wide gamut hdr-video for the evaluation of tone mapping operators and hdr-displays,” 2014. [Online]. Available: <http://spiedigitallibrary.org> *Cited in Sec.* 5.8.5
- [50] H. Gao, H. Yuan, Z. Wang, and S. Ji, “Pixel deconvolutional networks,” *arXiv preprint arXiv:1705.06820*, 2017. *Cited in Sec.* (document), 5.8.2, 5.18
- [51] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2414–2423. *Cited in Sec.* 5.3.4

- [52] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, “Deep joint demosaicking and denoising,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 191, 2016. *Cited in Sec. 5.1, 5.2.2*
- [53] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, “Deep bilateral learning for real-time image enhancement,” *ACM Trans. Graph.*, Jul. 2017. *Cited in Sec. 5.2.2*
- [54] S. Gidaris and N. Komodakis, “Detect, replace, refine: Deep structured prediction for pixel wise labeling,” in *CVPR*, 2017. *Cited in Sec. 3.1, 3.4.2*
- [55] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in *ICLR 2018*, 2018. *Cited in Sec. 1*
- [56] N. Girard and Y. Tarabalka, “End-to-end learning of polygons for remote sensing image classification,” in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 2083–2086. *Cited in Sec. 3.2*
- [57] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. *Cited in Sec. 1*
- [58] D. Gommelet, A. Roumy, C. Guillemot, M. Ropert, and J. L. Tanou, “Gradient-based tone mapping for rate-distortion optimized backward-compatible high dynamic range compression,” *IEEE Transactions on Image Processing*, pp. 5936–5949, Dec 2017. *Cited in Sec. 5.1*
- [59] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680. *Cited in Sec. 4.1, 4.6.1, 5.1, 5.2.3*
- [60] N. Greeshma, M. Baburaj, and S. N. George, “Reconstruction of cloud-contaminated satellite remote sensing images using kernel pca-based image modelling,” *Arabian Journal of Geosciences*, vol. 9, no. 3, p. 239, 2016. *Cited in Sec. 4.2.1*
- [61] P. Hanhart, M. Rerabek, and T. Ebrahimi, “Towards high dynamic range extensions of hevc: subjective evaluation of potential coding technologies,” *Applications of Digital Image Processing XXXVIII*, p. 95990G, 2015. *Cited in Sec. 5.6.2*
- [62] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, Dec 2011. *Cited in Sec. 4.2.2, 4.2, 4.3, ??, 4.8e, 4.6.2, 4.6.3*
- [63] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778. *Cited in Sec. 1, 3.4.2*
- [64] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006. *Cited in Sec. 1*
- [65] X. Hou, J. Duan, and G. Qiu, “Deep feature consistent deep image transformations: Downscaling, decolorization and HDR tone mapping,” *CoRR*, 2017. *Cited in Sec. 5.2.2*
- [66] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, “Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery,” *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, 2015. *Cited in Sec. 1, 2.1*
- [67] J. Hu, O. Gallo, K. Pulli, and X. Sun, “Hdr deghosting: How to deal with saturation?” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1163–1170. *Cited in Sec. 1*
- [68] B. Huang, Y. Li, X. Han, Y. Cui, W. Li, and R. Li, “Cloud removal from optical satellite imagery with sar imagery using sparse representation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 1046–1050, May 2015. *Cited in Sec. 4.1*

- [69] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962. *Cited in Sec.* 2.1
- [70] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015. *Cited in Sec.* 4.5.2, 5.5
- [71] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017. *Cited in Sec.* (document), 5.4, 5.2.3, 5.3.2, 5.8.2
- [72] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CVPR*, 2017. *Cited in Sec.* 4.1, 4.2.3, 4.5.2
- [73] H. Jiang, E. Learned-Miller, G. Larsson, M. Maire, and G. Shakhnarovich, "Self-supervised relative depth learning for urban scene understanding," 2017. *Cited in Sec.* 1
- [74] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, Jul 1997. *Cited in Sec.* 4.2, 4.3, ??, 4.8c, 4.6.2, 4.6.3
- [75] 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. *Cited in Sec.* 5.2.3, 5.3.1, 5.3.2, 5.3.4
- [76] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler, "Learning aerial image segmentation from online maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 11, pp. 6054–6068, 2017. *Cited in Sec.* 3.2
- [77] N. K. Kalantari and R. Ramamoorthi, "Deep high dynamic range imaging of dynamic scenes," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, vol. 36, no. 4, 2017. *Cited in Sec.* 5.2.2
- [78] H. J. Kelley, "Gradient theory of optimal flight paths," *Ars Journal*, vol. 30, no. 10, pp. 947–954, 1960. *Cited in Sec.* 1
- [79] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654. *Cited in Sec.* 1
- [80] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. *Cited in Sec.* 2.3.3, 3.4.3, 4.5.3, 5.5
- [81] G. Krawczyk. (2006) Mpi hdr video database. [Online]. Available: <http://resources.mpi-inf.mpg.de/hdr/video/> *Cited in Sec.* 5.8.5
- [82] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. *Cited in Sec.* 1, 5.1
- [83] R. Lanari, G. Fornaro, D. Riccio, M. Migliaccio, K. P. Papathanassiou, J. R. Moreira, M. Schwabisch, L. Dutra, G. Puglisi, G. Franceschetti *et al.*, "Generation of digital elevation models by using sir-c/x-sar multifrequency two-pass interferometry: the etna case study," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 5, pp. 1097–1114, 1996. *Cited in Sec.* 1
- [84] G. W. Larson, H. Rushmeier, and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," *IEEE Transactions on Visualization and Computer Graphics*, pp. 291–306, Oct. 1997. *Cited in Sec.* 5.2.1, 5.4.1, 5.1
- [85] R. Lasaponara and N. Masini, "Detection of archaeological crop marks by using satellite quickbird multispectral imagery," *Journal of archaeological science*, vol. 34, no. 2, pp. 214–221, 2007. *Cited in Sec.* 1
- [86] P. Ledda, A. Chalmers, T. Troscianko, and H. Seetzen, "Evaluation of tone mapping operators using a high dynamic range display," *ACM Transactions on Graphics (TOG)*, pp. 640–648, 2005. *Cited in Sec.* 5.1, 5.2.1, 5.4.1

- [87] 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,” *arXiv preprint arXiv:1609.04802*, 2016. *Cited in Sec.* (document), 5.1, 5.4, 5.2.3, 5.3.2
- [88] C. Li and M. Wand, “Precomputed real-time texture synthesis with markovian generative adversarial networks,” in *European Conference on Computer Vision*. Springer, 2016, pp. 702–716. *Cited in Sec.* (document), 5.1, 5.4, 5.3.2
- [89] H. Li, J. Peng, C. Tao, J. Chen, and M. Deng, “What do we learn by semantic scene understanding for remote sensing imagery in CNN framework?” *CoRR*, vol. abs/1705.07077, 2017. [Online]. Available: <http://arxiv.org/abs/1705.07077> *Cited in Sec.* 2.1
- [90] J. Li, X. Zhao, Y. Li, Q. Du, B. Xi, and J. Hu, “Classification of hyperspectral imagery using a new fully convolutional neural network,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 292–296, Feb 2018. *Cited in Sec.* 2.1
- [91] S. Liang, H. Fang, and M. Chen, “Atmospheric correction of landsat etm+ land surface imagery. i. methods,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 11, pp. 2490–2498, Nov 2001. *Cited in Sec.* 4.2.2
- [92] J. Liu, X. Wang, M. Chen, S. Liu, X. Zhou, Z. Shao, and P. Liu, “Thin cloud removal from single satellite images,” *Optics express*, vol. 22, no. 1, pp. 618–632, 2014. *Cited in Sec.* 4.3
- [93] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” *arXiv preprint arXiv:1703.00848*, 2017. *Cited in Sec.* 5.2.3
- [94] Y. Liu, Y. Liu, and L. Ding, “Scene classification based on two-stage deep feature fusion,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 183–186, Feb 2018. *Cited in Sec.* ??, ??
- [95] Z. Liu and B. R. Hunt, “A new approach to removing cloud cover from satellite imagery,” *Computer vision, graphics, and image processing*, vol. 25, no. 2, pp. 252–256, 1984. *Cited in Sec.* 4.3
- [96] J. Long, Z. Shi, W. Tang, and C. Zhang, “Single remote sensing image dehazing,” *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 1, pp. 59–63, Jan 2014. *Cited in Sec.* 4.2.2
- [97] L. Lorenzi, F. Melgani, and G. Mercier, “Inpainting strategies for reconstruction of missing data in vhr images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 5, pp. 914–918, Sept 2011. *Cited in Sec.* 4.2.2, 4.6.2
- [98] F. P. Luus, B. P. Salmon, F. Van den Bergh, and B. T. J. Maharaj, “Multiview deep learning for land-use classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2448–2452, 2015. *Cited in Sec.* 1
- [99] H. Lyu, H. Lu, and L. Mou, “Learning a transferable change rule from a recurrent neural network for land cover change detection,” *Remote Sensing*, vol. 8, no. 6, 2016. *Cited in Sec.* 2.1, 4.1
- [100] K. Ma, H. Yeganeh, K. Zeng, and Z. Wang, “High dynamic range image compression by optimizing tone mapped image quality index,” *IEEE Transactions on Image Processing*, vol. 24, Oct 2015. *Cited in Sec.* 5.1, 5.2.1, 5.3.1, 5.4.1
- [101] A. Maalouf, P. Carre, B. Augereau, and C. Fernandez-Maloigne, “A bandelet-based inpainting technique for clouds removal from remotely sensed images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2363–2371, July 2009. *Cited in Sec.* 4.2.2, 4.6.2
- [102] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models.” *Cited in Sec.* 5.8.1
- [103] E. Maggiori, G. Charpiat, Y. Tarabalka, and P. Alliez, “Recurrent neural networks to correct satellite image classification maps,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 4962–4971, Sept 2017. *Cited in Sec.* 3.1, 3.2

- [104] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “High-resolution aerial image labeling with convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 7092–7103, Dec 2017. *Cited in Sec.* 3.1
- [105] —, “Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017. *Cited in Sec.* (document), 3.1, 3.2, 3.4.1, ??, ??, 3.5.1, 4.1
- [106] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *CVPR 2015*, 2015, pp. 5188–5196. *Cited in Sec.* 2.1
- [107] K. Makantasis, K. Karantzas, A. Doulamis, and N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, 2015, pp. 4959–4962. *Cited in Sec.* 1
- [108] R. Mantiuk, R. Mantiuk, A. Tomaszewska, and W. Heidrich, “Color correction for tone mapping,” *Computer Graphics Forum*, 2009. *Cited in Sec.* (document), 5.16, 5.7.1
- [109] R. Mantiuk, K. Myszkowski, and H. P. Seidel, “A perceptual framework for contrast processing of high dynamic range images,” *ACM Trans. Appl. Percept.*, vol. 3, no. 3, Jul. 2006. *Cited in Sec.* 5.2.1, ??, 5.4.1, 5.1, 5.6, 5.6.1
- [110] R. Mantiuk, S. Daly, and L. Kerofsky, “Display adaptive tone mapping,” in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3. ACM, 2008, p. 68. *Cited in Sec.* 1
- [111] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017, pp. 2813–2821. *Cited in Sec.* 4.4, 5.5
- [112] D. Marmanis, K. Schindler, J. Wegner, S. Galliani, M. Datcu, and U. Stilla, “Classification with an edge: Improving semantic image segmentation with boundary detection,” vol. 135, 11 2016. *Cited in Sec.* 3.1
- [113] J. McCann and A. Rizzi, *The art and science of HDR imaging*, 01 2012. *Cited in Sec.* 5.1, 5.2.1
- [114] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014. *Cited in Sec.* 4.2.3, 5.1, 5.2.3, 5.3.2, 5.3.4
- [115] T. Miyato, S. ichi Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: A regularization method for supervised and semi-supervised learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 1979–1993, 2019. *Cited in Sec.* 1
- [116] V. Mnih and G. E. Hinton, “Learning to detect roads in high-resolution aerial images,” in *European Conference on Computer Vision*. Springer, 2010, pp. 210–223. *Cited in Sec.* 1
- [117] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013. [Online]. Available: <https://arxiv.org/pdf/1312.5602.pdf> *Cited in Sec.* 1
- [118] H. Z. Nafchi, A. Shahkolaei, R. F. Moghaddam, and M. Cheriet, “Fsim: A feature similarity index for tone-mapped images,” *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1026–1029, Aug 2015. *Cited in Sec.* 5.2.1
- [119] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814. *Cited in Sec.* 5.8.1
- [120] A. Odena, V. Dumoulin, and C. Olah, “Deconvolution and checkerboard artifacts,” *Distill*, vol. 1, no. 10, p. e3, 2016. *Cited in Sec.* (document), 5.8.2, 5.18
- [121] Onera, “Medusa toolbox,” <http://w3.onera.fr/medusa/downloading>, 2017. *Cited in Sec.* 4.5.1

- [122] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Is object localization for free? - weakly-supervised learning with convolutional neural networks,” in *CVPR*, June 2015, pp. 685–694. *Cited in Sec. 2.2.2*
- [123] A. Pardo and G. Sapiro, “Visualization of high dynamic range images,” *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 639–647, June 2003. *Cited in Sec. 5.1*
- [124] S. Parthasarathy and P. Sankaran, “An automated multi scale retinex with color restoration for image enhancement,” in *2012 National Conference on Communications (NCC)*, Feb 2012, pp. 1–5. *Cited in Sec. 4.2, 4.3, ??, 4.8c, 4.6.2, 4.6.3*
- [125] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” in *NIPS-W*, 2017. *Cited in Sec. 4.5.3, 5.5*
- [126] V. A. Patel, P. Shah, and S. Raman, “A generative adversarial network for tone mapping hdr images,” in *Computer Vision, Pattern Recognition, Image Processing, and Graphics*, Singapore, 2018, pp. 220–231. *Cited in Sec. 5.2.3, 5.4.1*
- [127] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, “Context encoders: Feature learning by inpainting,” 2016. *Cited in Sec. 4.2.3*
- [128] S. Pattanaik and H. Yee, “Adaptive gain control for high dynamic range image display,” in *Proceedings of the 18th Spring Conference on Computer Graphics*, ser. SCCG ’02. ACM, 2002, pp. 83–87. *Cited in Sec. 5.2.1, 5.4.1, 5.1, 5.6, 5.6.1*
- [129] K. Perlin, “Improving noise,” in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’02. New York, NY, USA: ACM, 2002, pp. 681–682. [Online]. Available: <http://doi.acm.org/10.1145/566570.566636> *Cited in Sec. 4.5.1*
- [130] Pfstools. (2007) Pfstools image database. [Online]. Available: <http://pfstools.sourceforge.net/hdr/gallery.html> *Cited in Sec. 5.8.5*
- [131] L. Poggio, A. Gimona, and I. Brown, “Spatio-temporal modis evi gap filling under cloud cover: An example in scotland,” *ISPRS journal of photogrammetry and remote sensing*, vol. 72, pp. 56–72, 2012. *Cited in Sec. 4.2.1*
- [132] M. Qin, F. Xie, W. Li, Z. Shi, and H. Zhang, “Dehazing for multispectral remote sensing images based on a convolutional neural network with the residual architecture,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 5, pp. 1645–1655, May 2018. *Cited in Sec. 4.2.1, 4.3, 4.6.2*
- [133] A. A. Rad, L. Meylan, P. Vandewalle, and S. Süsstrunk, “Multidimensional image enhancement from a set of unregistered and differently exposed images,” in *Computational Imaging V, San Jose, CA, USA, January 29-31, 2007*, 2007. [Online]. Available: <http://lcavwww.epfl.ch/alumni/meylan/> *Cited in Sec. 5.8.5*
- [134] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015. *Cited in Sec. 4.2.3*
- [135] P. Rakwatin, W. Takeuchi, and Y. Yasuoka, “Restoration of aqua modis band 6 using histogram matching and local least squares fitting,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 2, pp. 613–627, Feb 2009. *Cited in Sec. 4.2.1*
- [136] A. Rana, G. Valenzise, and F. Dufaux, “Evaluation of feature detection in HDR based imaging under changes in illumination conditions,” in *IEEE International Symposium on Multimedia, ISM 2015, Miami, USA, December, 2015*, 2015, pp. 289–294. *Cited in Sec. 5.8.5*
- [137] —, “Learning-based tone mapping operator for efficient image matching,” *IEEE Transactions on Multimedia*, pp. 1–1, 2018. *Cited in Sec. 1, 5.2.1*
- [138] K. Regmi and A. Borji, “Cross-view image synthesis using conditional gans,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. *Cited in Sec. 4.2.3*

- [139] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph.*, pp. 267–276, Jul. 2002. *Cited in Sec.* 5.4.1, 5.1, 5.6, 5.6.1
- [140] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99. *Cited in Sec.* 1
- [141] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *European Conference on Computer Vision*, 2016. *Cited in Sec.* 4.2, 4.3, ??, 4.8k, 4.6.2, 4.6.3
- [142] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241. *Cited in Sec.* 1, 5.8.2
- [143] R. E. Rossi, J. L. Dungan, and L. R. Beck, "Kriging in the shadows: geostatistical interpolation for remote sensing," *Remote Sensing of Environment*, vol. 49, no. 1, pp. 32–40, 1994. *Cited in Sec.* 4.2.2
- [144] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. *Cited in Sec.* 5.3.4
- [145] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Artificial intelligence and statistics*, 2009, pp. 448–455. *Cited in Sec.* 1
- [146] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 791â798. [Online]. Available: <https://doi.org/10.1145/1273496.1273596> *Cited in Sec.* 1
- [147] T. Sandhan and J. Y. Choi, "Simultaneous detection and removal of high altitude clouds from an image," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 4789–4798. *Cited in Sec.* 1, 4.2.2, 4.3, 4.6.2
- [148] C. Schlick, "An adaptive sampling technique for multidimensional integration by ray-tracing," in *Photorealistic Rendering in Computer Graphics*. Springer, 1994. *Cited in Sec.* 5.2.1, 5.3.1, 5.4.1, 5.1, 5.6
- [149] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017. *Cited in Sec.* 3.4.2
- [150] H. Shen, H. Li, Y. Qian, L. Zhang, and Q. Yuan, "An effective thin cloud removal procedure for visible remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 96, pp. 224–235, 2014. *Cited in Sec.* 4.1, 4.2.1
- [151] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. *Cited in Sec.* 5.3.4
- [152] P. Singh, Z. Kato, and J. Zerubia, "A multilayer markovian model for change detection in aerial image pairs with large time differences," in *2014 22nd International Conference on Pattern Recognition*, Aug 2014, pp. 924–929. *Cited in Sec.* 4.1
- [153] P. Singh and N. Komodakis, "Effective building extraction by learning to detect and correct erroneous labels in segmentation mask," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2018. *Cited in Sec.* 4.1, 4.6.4
- [154] M. Song, D. Tao, C. Chen, J. Bu, J. Luo, and C. Zhang, "Probabilistic exposure fusion," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 341–357, 2012. *Cited in Sec.* 1
- [155] J.-P. Tarel, N. Hautière, A. Cord, D. Gruyer, and H. Halmaoui, "Improved visibility of road scene images under heterogeneous fog," in *Proceedings of IEEE Intelligent Vehicle Symposium (IV'2010)*, San Diego, California, USA, 2010, pp. 478–485, <http://perso.lcpc.fr/tarel.jean-philippe/publis/iv10.html>. *Cited in Sec.* 4.2, 4.3, ??, 4.8g, 4.6.2, 4.6.3

- [156] J.-P. Tarel and N. Hautière, “Fast visibility restoration from a single color or gray level image,” in *Proceedings of IEEE International Conference on Computer Vision (ICCV’09)*, Kyoto, Japan, 2009, pp. 2201–2208, <http://perso.lcpc.fr/tarel.jean-philippe/publis/iccv09.html>. Cited in Sec. 4.2, 4.3, ??, 4.8g, 4.6.2, 4.6.3
- [157] O. Tasar, E. Maggiori, P. Alliez, and Y. Tarabalka, “Polygonization of binary classification maps using mesh approximation with right angle regularity,” in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 6404–6407. Cited in Sec. 3.2
- [158] C. J. Tucker, D. M. Grant, and J. D. Dykstra, “Nasa’s global orthorectified landsat data set,” *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 3, pp. 313–322, 2004. Cited in Sec. 1
- [159] J. Tumblin, J. K. Hodgins, and B. K. Guenter, “Two methods for display of high contrast images,” *ACM Trans. Graph.*, pp. 56–94, Jan. 1999. Cited in Sec. 5.2.1, 5.4.1, 5.1
- [160] H. Tung, A. Harley, W. Seto, and K. Fragkiadaki, “Adversarial inversion: Inverse graphics with adversarial priors,” *arXiv preprint arXiv:1705.11166*, 2017. Cited in Sec. 5.2.3
- [161] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *CoRR*, vol. abs/1607.08022, 2016. [Online]. Available: <http://arxiv.org/abs/1607.08022> Cited in Sec. 4.5.2, 5.5, 5.8.1
- [162] M. Vakalopoulou, K. Karantza, N. Komodakis, and N. Paragios, “Building detection in very high resolution multispectral data with deep learning features,” in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, 2015, pp. 1873–1876. Cited in Sec. 1, 3.2
- [163] F. Van der Meer, “Remote-sensing image analysis and geostatistics,” *International Journal of Remote Sensing*, vol. 33, no. 18, pp. 5644–5676, 2012. Cited in Sec. 4.2.2
- [164] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010. Cited in Sec. 1
- [165] M. Volpi and D. Tuia, “Dense semantic labeling of subdecimeter resolution images with convolutional neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 881–893, Feb 2017. Cited in Sec. 2.1
- [166] M. Wan and X. Li, “Removing thin cloud on single remote sensing image based on swf,” in *Online Analysis and Computing Science (ICOACS), IEEE International Conference of*. IEEE, 2016, pp. 397–400. Cited in Sec. 4.2.1
- [167] P. Wang, H. Zhang, and V. M. Patel, “Sar image despeckling using a convolutional neural network,” *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1763–1767, 2017. Cited in Sec. 1
- [168] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” *arXiv preprint arXiv:1711.11585*, 2017. Cited in Sec. 5.2.3, 5.3.3
- [169] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004. Cited in Sec. 4.6.2
- [170] H. Wu, B. Liu, W. Su, W. Zhang, and J. Sun, “Deep filter banks for land-use scene classification,” *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1895–1899, Dec 2016. Cited in Sec. ??
- [171] —, “Hierarchical coding vectors for scene level land-use classification,” *Remote Sensing*, vol. 8, no. 5, 2016. Cited in Sec. ??
- [172] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang, “AID: A benchmark dataset for performance evaluation of aerial scene classification,” *CoRR*, vol. abs/1608.05167, 2016. Cited in Sec. 2.1, ??, ??, 2.3.1, ??, ??, ??, 2.3.2, ??, ??, ??, 2.3.3, 2.4

- [173] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maître, “Structural high-resolution satellite image indexing,” 2010. *Cited in Sec. 2.1, ??, 2.3.1*
- [174] F. Xiao, J. M. DiCarlo, P. B. Catrysse, and B. A. Wandell, “High dynamic range imaging of natural scenes,” in *In Tenth Color Imaging Conference: Color Science, Systems, and Applications*, 2002. *Cited in Sec. 5.8.5*
- [175] J. Xie, L. Xu, and E. Chen, “Image denoising and inpainting with deep neural networks,” in *Advances in neural information processing systems*, 2012, pp. 341–349. *Cited in Sec. 1*
- [176] M. Xu, X. Jia, M. Pickering, and A. J. Plaza, “Cloud removal based on sparse representation via multitemporal dictionary learning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2998–3006, May 2016. *Cited in Sec. 4.1*
- [177] M. Xu, M. Pickering, A. J. Plaza, and X. Jia, “Thin cloud removal based on signal transmission principles and spectral mixture analysis,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1659–1669, March 2016. *Cited in Sec. 4.1*
- [178] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS ’10. ACM, 2010, pp. 270–279. *Cited in Sec. 2.1, ??, 2.3.1*
- [179] H. Yeganeh and Z. Wang, “Objective quality assessment of tone-mapped images,” *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 657–667, Feb 2013. *Cited in Sec. 5.2.1, 5.4.1*
- [180] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” in *CVPR*, vol. 2, no. 3, 2017, p. 4. *Cited in Sec. 1, 5.2.3*
- [181] Y. Yu and F. Liu, “Aerial scene classification via multilevel fusion based on deep convolutional neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 287–291, Feb 2018. *Cited in Sec. 1*
- [182] C. Zeng, H. Shen, and L. Zhang, “Recovering missing pixels for landsat etm+ slc-off imagery using multi-temporal regression analysis and a regularization method,” *Remote Sensing of Environment*, vol. 131, pp. 182–194, 2013. *Cited in Sec. 4.2.1*
- [183] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” *arXiv preprint arXiv:1701.05957*, 2017. *Cited in Sec. 4.2.3*
- [184] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *European Conference on Computer Vision*. Springer, 2016, pp. 649–666. *Cited in Sec. 1*
- [185] Y. Zhang, B. Guindon, and J. Cihlar, “An image transform to characterize and compensate for spatial variations in thin cloud contamination of landsat images,” *Remote Sensing of Environment*, vol. 82, no. 2-3, pp. 173–187, 2002. *Cited in Sec. 4.2.2*
- [186] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” *CVPR*, 2016. *Cited in Sec. 1*
- [187] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929. *Cited in Sec. 2.2.2*
- [188] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” *arXiv preprint arXiv:1703.10593*, 2017. *Cited in Sec. 4.1, 4.2.3, 4.5.2, 4.5.3, 5.2.3, 5.3.2*
- [189] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, “Generative adversarial networks for hyperspectral image classification,” *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–18, 2018. *Cited in Sec. 4.2.3*

- [190] X. Zhu, F. Gao, D. Liu, and J. Chen, "A modified neighborhood similar pixel interpolator approach for removing thick clouds in landsat images," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 521–525, May 2012. *Cited in Sec. 4.2.1*
- [191] H. Zimmer, A. Bruhn, and J. Weickert, "Freehand hdr imaging of moving scenes with simultaneous resolution enhancement," in *Computer Graphics Forum*, vol. 30, no. 2. Wiley Online Library, 2011, pp. 405–414. *Cited in Sec. 1*
- [192] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep Learning Based Feature Selection for Remote Sensing Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp. 2321–2325, Nov. 2015. *Cited in Sec. 2.1, ??, 2.3.1*