Improving CNN classification performance for small datasets

Midas Amersfoort

Layout: typeset by the author using LaTeX.
Cover illustration: Unknown artist

Improving CNN classification performance for small datasets

Post-natural disaster building damage assessment from high-resolution satellite imagery

Midas Amersfoort
11857412

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Dhr. dr. A. Visser

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

June 26th, 2020

Abstract

As demonstrated in recent years, Convolutional Neural Networks (CNNs) are powerful tools for image recognition tasks. A major contributing factor to this is their translation invariance. This property allows them to recognize objects – or rather the defining features of objects – regardless of their location in the image. However, whereas objects may appear anywhere in an image, their orientation should not change. This is because CNNs are not inherently rotation invariant. For most types of imagery, this has no significant implication, as most objects naturally have similar orientations for the majority of the time. However, with images taken from a satellite, objects can take on any orientation due to the satellite's sensors being able to take on any orientation. This can pose problems when dealing with raw satellite imagery. Another drawback of traditional CNNs is that they are not illumination invariant. Therefore, training a CNN under varying types of lighting requires additional training data, as it needs to learn the features for every type of illumination separately. To deal with these problems, this thesis proposes a combination of two methods for improving the building damage assessment performance of a CNN with small datasets of high-resolution satellite imagery. The first proposed method is to incorporate rotational equivariance into the CNN using Group equivariant Convolutional Neural Networks (G-CNNs). Secondly, a method for incorporating illumination invariance is proposed. The findings in this research demonstrate that a G-CNNs can provide a considerable increase in performance with smaller datasets when compared to a regular CNN. However, the implementation of illumination invariance does not yield a similar convincing uplift in performance.

Contents

Acknowledgements

I would like to thank my supervisor Arnoud Visser for assisting me throughout this project and providing me with valuable weekly feedback.

Furthermore, I would also like to thank the *Maxar Open Data Program*, *xBD* and the people behind the *xView2* challenge for their data.

And finally, I am thankful to my family for their support and encouragement.

1 Introduction

Every year natural disasters are responsible for many thousands of deaths throughout the world. Some of these deaths are a direct result of the disasters themselves, but another great portion are due to the aftermath they leave behind. When such a disaster strikes, quick and accurate situational information is viral for an effective response. However, before emergency aid can be provided to the affected area, the location and the severity of damage should be known. And with traditional boots-on-the-ground damage assessment methods being difficult, dangerous and slow, this has resulted in ineffective use of emergency aid, consequently leading to many unnecessary lives lost. Due to these limitations, efforts have been made to enable the assessment of damage through alternative methods. One such method is the assessment of building damage from satellite imagery. As satellite imagery is taken on a regular basis, changes in the earth's landscape can be derived from them. This, combined with recent developments in AI – in particular with regards to Convolutional Neural Networks (CNNs) – has enabled the possibility of automating the process of building damage assessment from satellite imagery. This would eliminate the risks involved with traditional damage assessment methods, in addition to providing a significant speed up to the entire process.

However, although these methods show great potential, one major limitation comes to light: There is a lack of large, high-resolution annotated datasets of satellite imagery. Moreover, if there is such a dataset available, often this is only for specific locations. Furthermore, although models have been trained on such datasets, which performed reasonably well, there is a lot of variety in the footprint concerning the materials used and the overall look of buildings throughout different regions in the world. Therefore, a model trained in one region may perform poorly in others, as it does not generalize well. As a result, these models are not deployed in regions other than those that were in the initial dataset.

With this limitation in mind, a combination of two methods is proposed to improve the performance of a CNN on small datasets: incorporating illumination invariance and rotational equivariance into the CNN. Integrating these properties into the CNN is expected to improve the quality and robustness of detected features in images, therefore improving the CNN's generalizing capability. Improving the CNN's performance for small datasets would reduce the amount of work that has to be put into creating huge annotated datasets. Rather, a small amount of training data could provide the CNN with enough data to achieve sufficient results. Therefore, – coming back to the limitation of building damage assessment

CNNs not generalizing well to previously unseen regions/building footprints – more effort could be put into expanding the dataset to other regions. This would open up the deployment of building damage assessment from satellite imagery to more regions. In an effort of realizing this, the research question thus becomes: *Can we improve a CNN's building damage assessment performance on small datasets of high-resolution satellite imagery by incorporating illumination invariance and rotational equivariance?*

Related work Several implementations for attaining illumination invariance have been proposed throughout the years. Methods such as utilizing depth information have demonstrated robustness against shadows, different weather conditions and changes in illumination [1] [2]. Additionally, different color spaces [3] [4] [5] have been proposed. Other works include camera specific illumination invariant spaces [6] [7]. However, achieving illumination invariance in satellite imagery has not been that thoroughly researched.

Regarding rotation robustness of CNNs, a considerable amount of literature is available about invariant representations. One of the most well known examples is SIFT [8], a local descriptor capable of extracting scale- and rotation-invariant features. Unfortunately, as it is a local descriptor, it fails to represent whole objects in classification tasks. Another widely used method to achieve rotational invariance in remote sensing image classification is the HOG [9]. The HOG can be seen as something of an extension to SIFT, the main difference being that it is computed over localized portions of an image. While the HOG can improve performance in some cases, it falls apart in more complex scenes. Double-Net [10] has demonstrated a rotation-invariant feature detection in satellite imagery. This was achieved by incorporating multiple channels with shared weights into the CNN, where each channel refers to a specific rotational direction. However, convergence speeds were slow.

Whereas in some cases invariance may be desired, equivariance is often more useful as the ability to determine the feature's spatial configuration is retained. Therefore, numerous other works have addresses the issue of learning equivariant representations. E.g. equivariant descriptors [11], equivariant Bolzmann machines [12], equivariant filtering [13] and equivariant deep symmetry networks [14].

2 Theoretical foundation

2.1 Illumination invariant imagery

In [15], a method for exploiting spectral properties of a camera's sensor to infer physical quantities about a scene as proposed. I.e. images are processed to be illumination invariant. This method is aimed at reducing the effects of changes in illumination and the influence of shadows that are cast as a result of directional lighting. In outdoor environments, variables such as lighting can significantly alter the appearance of a scene, in turn complicating image recognition tasks. This is because conventional CNNs are not inherently illumination invariant. Therefore, resolving issues regarding change in illumination could be highly beneficial for the performance of a CNN.

As established by multiple experiments [16] [17], the visible spectrum of natural lighting closely follows the Planckian locus. As such, following [18], the illumination spectrum can be estimated as a Planckian source:

$$log(R_i) = log(GI) + log(2hc^2\lambda_i^{-5}S_i) - \frac{hc}{k_B T \lambda_i} \tag{2.1}$$

where h is Planck's constant, c is the speed of light, k_B is the Boltzmann constant and T is the correlated color temperature (CCT) of the black-body source. Then, if the peak spectral responses at three different wavelengths λ_i (e.g. red, green and blue) are known, [15] proposes that equation 2.1 can be reduced to:

$$I = log(R_2) - \alpha \cdot log(R_1) - (1 - \alpha) \cdot log(R_3) \tag{2.2}$$

where I is a 1D color space and R_1, R_2 and R_3 correspond to the ordered peak spectral sensitivities $\lambda_1 < \lambda_2 < \lambda_3$. The 1D color space I will be invariant to the CCT if α satisfies equation 2.3 [15]. Thus, achieving illumination invariance.

$$\frac{1}{\lambda_2} = \frac{\alpha}{\lambda_1} + \frac{(1 - \alpha)}{\lambda_3} \tag{2.3}$$

2.2 Group equivariant Convolutional Neural Networks

In a paper by Taco S. Cohen *et al* [19], the notion of *Group equivariant Convolutional Neural Networks* (G-CNN) is introduced. Its improvement as opposed to a regular CNN is the inclusion of an additional layer: the G-convolution layer. The main purpose of this layer is to reduce sample complexity by exploiting symmetries. Moreover, the G-convolution layer shares its weights to a much higher degree than a normal convolution layer. With that, its expressive capacity is increased without an increase in the amount of parameters. To get a more in depth understanding of how exactly this works, the fundamentals of the G-convolution layer are explained below.

2.2.1 Symmetry groups

As mentioned, the G-convolution layer exploits symmetries. A symmetry of an object is defined as a transformations that, if applied, leaves the object invariant. To evoke a more intuitive understanding of what this means we will have a look at an example of a symmetry: the flip transformation. Let us take an image $\mathbb{R}^2$. If we apply the flip transformation we get:

$$-\mathbb{R}^2 = \{(-x, -y) | (x, y) \in \mathbb{R}^2\} = \mathbb{R}^2 \tag{2.4}$$

As can be seen from the equation above, the flip transformation leaves the image $\mathbb{R}^2$ invariant. Thus, the flip transformation is a symmetry of the image $\mathbb{R}^2$. Additionally, following from the fact that all symmetry transformations leave the object invariant, a combination of symmetry transformations results in another symmetry transformation. Using that same reasoning, it is not hard to see that the inverse of a symmetry also is a symmetry. A set of such symmetry transformations is known as a symmetry group.

2.2.2 The group p4(m)

Regarding symmetry groups, two that are used for the G-CNN are the group p4, and group p4m. The group p4 contains the set of all combinations of translations and rotations of 90 degrees around any center of rotation in an image. This group can be expressed in terms of three parameters: r, t_1 and t_2, which in matrix form is denoted as:

$$g(r, t_1, t_2) = \begin{bmatrix} cos(r\pi/2) & -sin(r\pi/2) & t_1 \\ sin(r\pi/2 & cos(r\pi/2 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

where $0 \leq r < 4$ and $(t_1, t_2) \in \mathbb{R}^2$. Applying this symmetric transformation to any point $\vec{x} \in \mathbb{R}^2$ is then done by multiplying matrix (2.5) to the homogeneous coordinate vector of that point $\vec{x}$:

$$g(r, t_1, t_2)\vec{x} = \begin{bmatrix} cos(r\pi/2) & -sin(r\pi/2) & t_1 \\ sin(r\pi/2 & cos(r\pi/2 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.6}$$

The group p4m is an extension of the group p4, with the addition of mirror reflections: mirrors axes both perpendicular and parallel to the main axis. The matrix representation of this group can be written as:

$$g(m, r, t_1, t_2) = \begin{bmatrix} (-1)^m cos(r\pi/2) & -(-1)^m - sin(r\pi/2) & t_1 \\ sin(r\pi/2 & cos(r\pi/2 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

where m $\in \{0,1\}$, $0 \leq r < 4$ and $(t_1, t_2) \in \mathbb{R}^2$. Applying this symmetric transformation to any point $\vec{x} \in \mathbb{R}^2$ naturally goes the same as for (2.6), by multiplying (2.7) with the homogeneous coordinate vector of that point $\vec{x}$:

$$g(m, r, t_1, t_2)\vec{x} = \begin{bmatrix} (-1)^m cos(r\pi/2) & -(-1)^m - sin(r\pi/2) & t_1 \\ sin(r\pi/2 & cos(r\pi/2 & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.8}$$

2.2.3 Exploiting symmetries to achieve group equivariance

The result of applying filters to an input image in a CNN is captured by feature maps. As an example, Figure 2.1 (left) shows a group p4 feature map. The four patches are associated with the four 90 degree rotations that are incorporated in the group p4. Then, if we apply a rotation of 90 degrees ($r = 1$) on this feature map, each patch follows the direction of its red arrow and is rotated 90 degrees. The resulting feature map after this transformation can be seen in Figure 2.1 (right). This is the exact transformation a feature map in a G-CNN with a group p4 layer would go through if the input image would be rotated in the same 90 degree rotation.

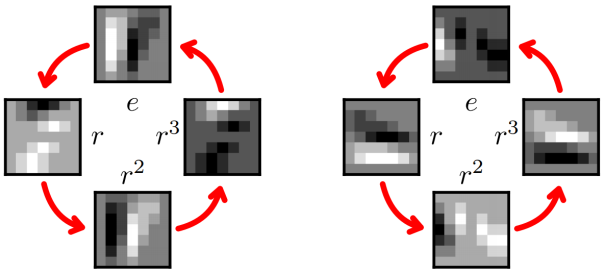


Figure 2.1: A group p4 feature map for a fourfold rotations r. Courtesy [19]

Elaborating on this, the group p4m feature map also includes mirror reflections. This feature map is illustrated in Figure 2.2 (left), where the 8 different patches represent the four 90 degree rotations r and their respective mirror reflections m. If we were to apply the same 90 degree rotation to the p4m feature map, each patch would again follow its red arrow and be rotated 90 degrees. The result of this transformation is shown in Figure 2.2 (right).

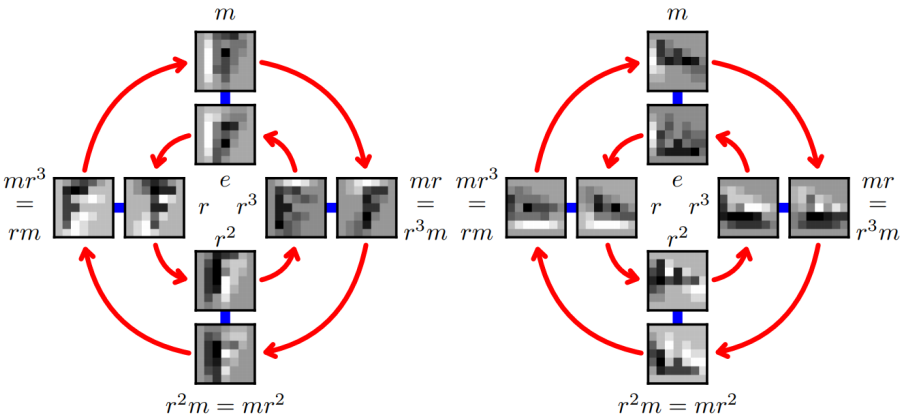


Figure 2.2: A group p4m feature map for a fourfold rotations r and their mirror reflections m. Courtesy [19]

The implications of using these symmetry groups become evident in Figure 2.3. Here, two images are fed into a two layer G-CNN [20] that uses the group p4. One is the original image, the other is its rotated duplicate. The $\mathbb{Z}^2 \rightarrow$ p4 convolution is performed by correlating the input image with a fourfold of 90 degree rotated instances of the kernel. Then, the result is correlated with the p4-kernel, which is again rotated fourfold. Lastly, average pooling over all orientations in the feature map is performed, resulting in the output. When comparing both outputs in the figure, the pooled output of the rotated input image is the exact 90 degree rotated duplicate of the original image's pooled output. Therefore, this network is

demonstrated to be equivariant to all 90 degree rotations. One may verify that this is the case for all transformations belonging to the group p4. Were we to extend this network to a p4m network, it would also be equivariant to all transformations belonging to that group.

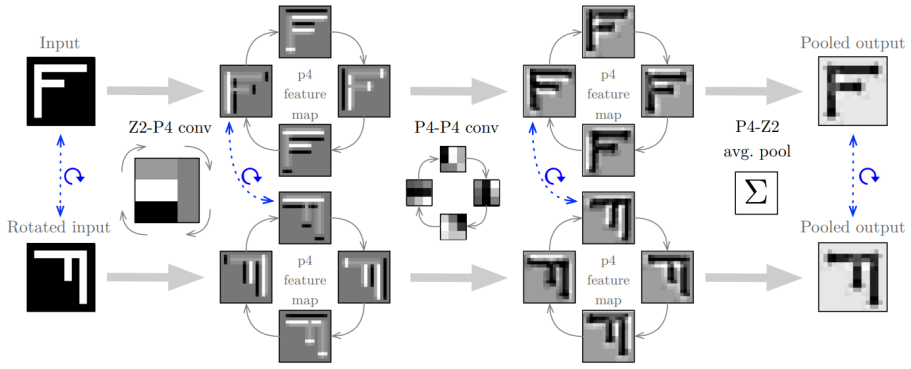


Figure 2.3: Demonstrates the rotation equivariant property of a G-CNN for p4. Courtesy [20]

3 The xBD dataset

The dataset used for training the network in this project was the xBD dataset [21]. At the moment of writing, this was the largest dataset with high-resolution annotated satellite imagery. Additionally, it was specifically designed for advancing building damage assessment methods, with the aim of improving post-disaster humanitarian assistance and disaster recovery. This made the xBD dataset very well suited for this project. It provided both pre- and post-disaster (RGB) satellite imagery from six different disaster types: earthquakes, tsunami's, floods, volcanic eruptions, wildfire and wind. The target resolution for all imagery in the dataset was 0.8m.

3.1 Full dataset

The full xBD dataset contained satellite imagery of nineteen separate disaster events. It should be noted that the amount of annotated buildings available from each disaster event was not equally distributed. Figure 3.1 shows the distribution of annotated buildings, specified in polygons, from each disaster event.

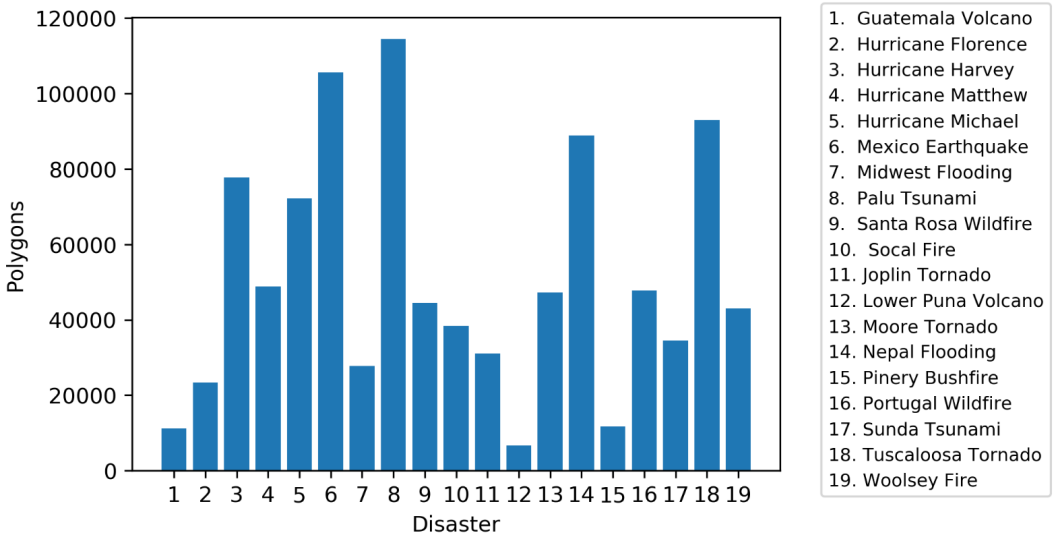


Figure 3.1: The number of annotated polygons in the xBD dataset per disaster event.

As mentioned, the xBD dataset provided both pre- and post-disaster satellite imagery. The reason for this being that it enabled the detection of change between the two images, which would be the result of the disaster that struck. An example pair of such pre- and post-disaster imagery can be seen in Figure 3.2a versus 3.2b. Annotation of the imagery in the xBD dataset was provided in the form of a JSON file for each individual image. These JSON files contained the outlines (polygons) of each building in the image, along with its corresponding damage classification label. A visualization of these building damage labels can be seen in in Figure 3.2c.

(a) Pre-disaster (b) Post-disaster (c) Building damage labels

Figure 3.2: A pair of pre- and post-disaster images with corresponding post-disaster building damage labels (green: no damage, red: destroyed).

However, whereas Figure 3.2c only shows two distinct damage classification labels, there are a total of four different damage classification labels throughout the whole dataset: 'no damage', 'minor damage', 'major damage' and 'destroyed'.

3.2 Dataset subselection

The aim of this project was to improve a CNN's performance on small datasets. Therefore, as the xBD was a relatively large dataset, the first step was to reduce the size of the dataset. This was done by making a subselection of the full xBD dataset. For this subselection, the idea was to pick a single disaster type to focus on. Specifically, the chosen disaster was hurricane Michael: a very powerful tropical cyclone that struck the contiguous United States in 2018 (the exact states contained in this dataset were not specified by xBD [21]). Making this subselection reduced the size of the dataset to 45372 building polygons, which is about one tenth of the full dataset. The reason for not choosing the disaster with the smallest amount of annotated polygons will become clear in Section 5, where the performance of our network was measured for different training sizes. While exploring all imagery belonging to the hurricane Michael disaster, it became evident that some buildings were given the label: 'unclassified'. After removing these occurrences, the distribution of damage classification labels was as shown in Figure 3.3.

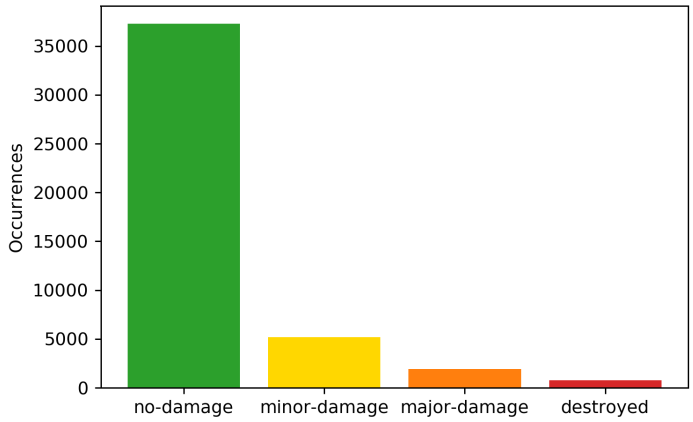


Figure 3.3: The distribution of damage classification labels for the hurricane Michael disaster.

4 Methodology

4.1 Extracting buildings from satellite imagery

The objective of our G-CNN was to correctly assess building damage from satellite imagery. This image classification problem had to be performed on a per building basis. However, each image in the xBD dataset spanned an area of over 800m^2. Therefore, the individual buildings first had to be extracted from these images. As mentioned in Section 3.1, all imagery in the xBD dataset came with a corresponding JSON file, which contained the polygons of all buildings in those images. These polygons were used to determine the location of buildings in each image. However, a simple cutout of each building polygon was not desired. I.e. the G-CNN proposed in Section 4.4 would only accept square images of a fixed dimension. Therefore, due to variations in the scale of buildings, building cutouts had to be resized to meet these requirements. The requirement of images needing to be square was satisfied by making a square cutout of each building around its polygon. A margin of 20 pixels was used for this cutout. This margin's purpose was to provide some additional situational context to each image. An example of this process's output can be seen in Figure 4.1. The second requirement: images needing to be of a fixed dimension, is discussed in Section 4.4.

Figure 4.1: Extracted building images from satellite imagery

4.2 Calculating the illumination invariant color space

With all images of buildings obtained, the illumination invariant color space had to be calculated. As described in Section 2.1, the illumination invariant color space could be computed if the peak spectral responses of an image's camera were known. Fortunately, for every image in the xBD dataset, the name of the satellite that captured the image was provided. All images were taken by either the GeoEye-1 or the WorldView-2 satellite. Their respective peak spectral responses to the red, green and blue band could be found in [22]. With this knowledge, the peak responses could be substituted into Equation 2.3. Solving for α yielded a value of 0.54 for both satellite's sensors. With α known, the illumination invariant color space could be computed (Equation 2.2) for all building images. After calculation, the 1D illumination invariant color space was added as a fourth channel to each original RGB image. Several example outputs of the illumination invariant color space can be seen in Figure 4.2 (bottom row).

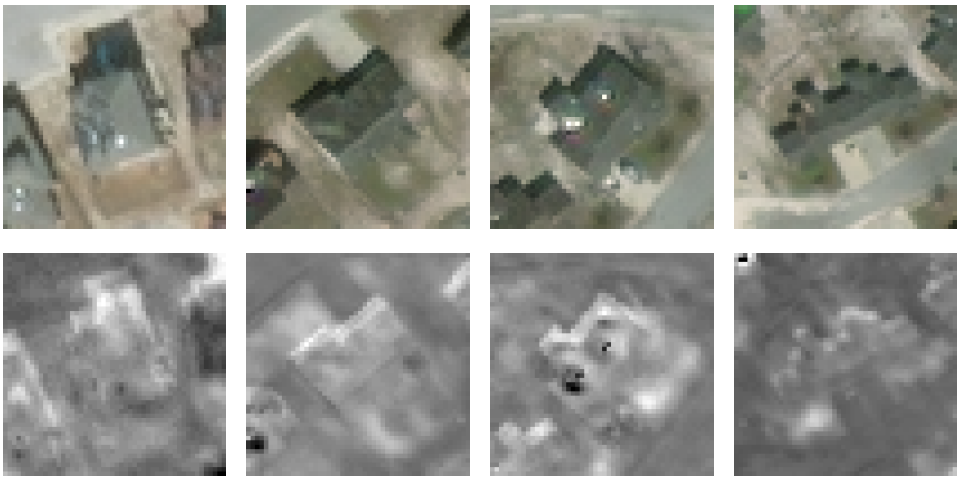


Figure 4.2: The illumination invariant color space (bottom) calculated for the images in Figure 4.1 (top).

4.3 Data augmentation

After obtaining all images of buildings and their illumination invariant color space, a split was made between the training, validation and test set. For this split, a ratio of 0.64, 0.16, 0.2 respectively was used. Then, data augmentation was performed on the training set to artificially enlarge it and to create more diversity. The image manipulations used for data augmentation were random zooms, adding Gaussian noise, adding Gaussian blur, introducing color manipulations (brightness, contrast, saturation, hue) and applying random affine transformations, where the image could be rotated up to 360 degrees and sheared up to 30 degrees. Apart from enlarging the dataset and creating more diversity, a driving factor behind performing data augmentation was to rectify the severe class imbalances (see Figure 3.3). Therefore, data augmentation was performed to a degree that was proportional to the size of each class. I.e. as the class 'no damage' contained the most images, only one augmentation was performed on each image belonging to this class. Furthermore, each image in the 'minor damage' class received seven augmentations, images in the class 'major damage' 19 and images in the class 'destroyed' 25. The class distribution after performing these augmentations can be seen in Figure 4.3. The validation and test set received no data augmentation (see Appendix A).

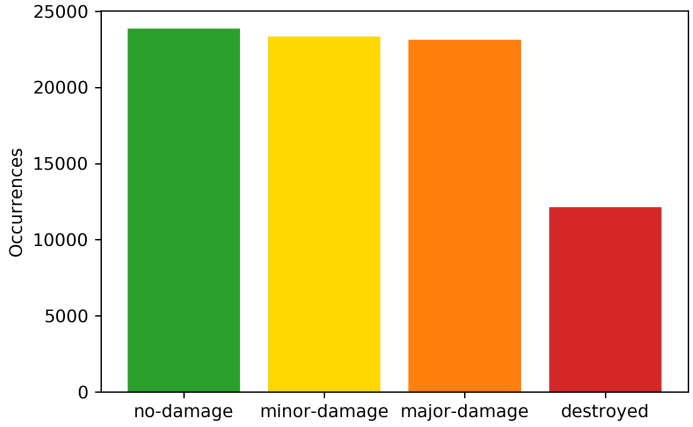


Figure 4.3: The distribution of training samples for each of the four damage types after data augmentation.

4.4 Implementing rotational equivariance

With the necessary preprocessing steps and data augmentations performed, rotational equivariance could be implemented. In Section 2.2, the principle of G-CNNs was introduced. The main difference with traditional CNNs being their usage of G-convolution layers, which exploit symmetries to achieve group-equivariance. Incorporation of the group p4 or group p4m G-convolution layers into a CNN was demonstrated to render it equivariant to all transformations belonging to their respective group. As the rotation transformation – which rotates by multiples of 90 degrees – belonged to both groups, the aim was to incorporate the G-convolution layers into our own G-CNN to achieve rotational equivariance.

A DenseNet [23] architecture, based on the one proposed by B.S. Veeling *et al* [20], was used for our G-CNN. Differing from B.S. Veeling *et al*, fewer dense blocks were used – as the images we worked with were of a lower resolution – in addition to an alternative activation function, i.e. the Softmax activation function. The particular reason for using the DenseNet architecture was due to its combination with the p4 and p4m G-convolution layers being previously demonstrated to outperform traditional CNNs on binary classification tasks [20].

G-CNN architecture

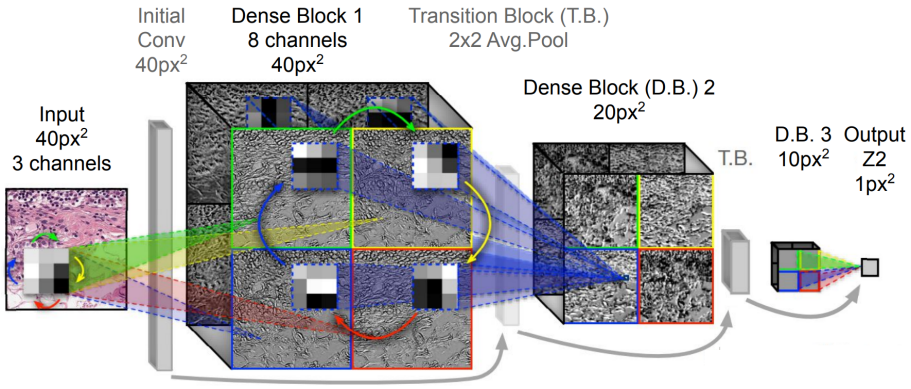


Figure 4.4: The equivariant DenseNet architecture for p4. Courtesy [20].

Figure 4.4 illustrates the equivariant DenseNet architecture for p4 (Figure 2.1). The p4m (Figure 2.2) version requires a straightforward extension, but was not displayed for the purpose of readability. As shown, the number of input channels reads three. However, when including the illumination invariant color space, naturally this number increases to four. As illustrated in Figure 4.4, the architecture alternates between Dense Blocks and Transition Blocks. The Dense Block's

layers use the stacked previous layers as their input and the Transition Blocks consist of a 1×1 convolutional layer and a 2×2 strided Average Pool. To achieve group-equivariance in the whole model, the convolution layers are replaced with G-convolution layers [19]. Then, as proposed by Taco S. Cohen *et al* [19], to achieve group-equivariance in the batch normalization layers, moments are combined per group feature map as opposed to per spatial feature map. Lastly, the output layer is preceded by a group-pooling layer with a subsequent Softmax activation function.

It should be noted that the actual architecture used for testing utilized the extended p4m version rather than the one depicted in Figure 4.4. Furthermore, as the input of the model demands a fixed input size of 40×40 pixels, all images were resized to these dimensions using a Lanczos filter [24] prior to being fed into the model.

5 Results

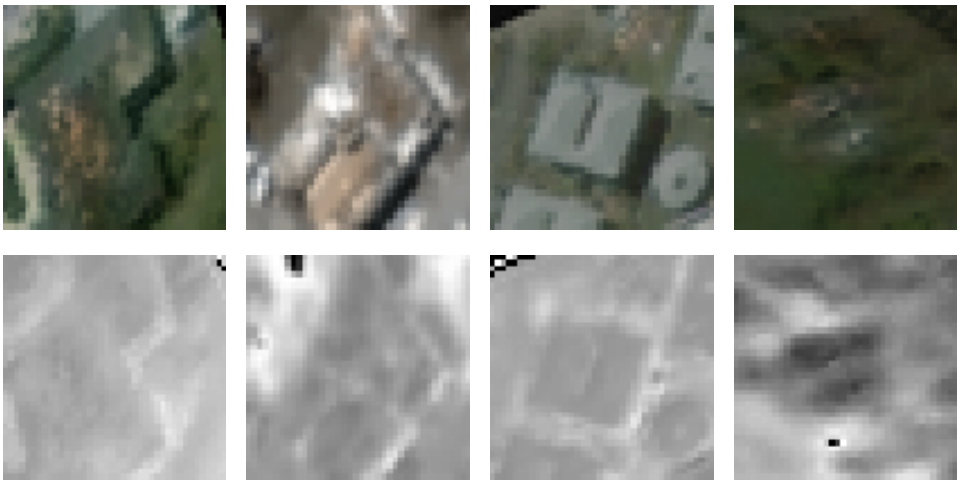
In this section, four different configurations of CNN models were compared: the traditional CNN, the illumination invariant CNN, the G-CNN and illumination invariant G-CNN. Figure 4.4 illustrates the G-CNN's architecture. Furthermore, the architecture that constituted the traditional CNN was based on the DenseNet architecture, similar to that of the G-CNN. However, with the difference of lacking G-convolution layers (Section 2.2). Instead, it used regular convolution layers. Therefore, the traditional CNN was not equivariant to the group p4m (Figure 2.2), whereas the G-CNN was. Lastly, each of these two models was regarded to be illumination invariant if the illumination invariant color space (see Section 2.1) was provided as an additional fourth input channel to the model.

5.1 Model predictions

To demonstrate the differences between the four models, several post-disaster damage assessment predictions for images from the test set were reviewed. For all models, the exact same training, validation and test set was used, measuring a size of 4800, 1200 and 1500 respectively. This was done to ensure a level playing field for all models. Any variations in outcome could therefore be attributed to differences between the models themselves.

5.1.1 CNN vs illumination invariant CNN

Here, a conventional CNN was compared to a CNN with the included illumination invariant color space. Figure 5.1 shows four sample images from the test set. These four images were given as an input to their respective models, after which a damage assessment prediction was made. The output of these predictions is shown in Table 5.1 and Table 5.2.



(a) (b) (c) (d)

Figure 5.1: Four sample images from the test set (top row) with their respective illumination invariant color space (bottom row).

Naturally, the illumination invariant color space images in the bottom row of Figure 5.1 were not provided as an input to the conventional CNN, as it is not illumination invariant.

Table 5.1: CNN building damage assessment predictions for the images in Figure 5.1. Illustrates the probability of each image belonging to a certain class where the class with the highest probability is chosen as prediction (Underlined = correct classification, green = correctly classified, red = falsely classified).

	No-damage	Minor-damage	Major-damage	Destroyed
(a)	0.04	0.40	0.41	0.15
(b)	0.02	0.09	0.38	0.51
(c)	0.18	0.39	0.40	0.03
(d)	0.25	0.23	0.15	0.37

As shown in Table 5.1, three out of the four predictions made by the conventional CNN were false. The samples in Fig. 5.1a and 5.1c were classified as majorly damaged instead of minorly damaged, while Fig. 5.1b was classified as destroyed instead of majorly damaged. Only Fig. 5.1d was correctly classified in the category destroyed. However, whereas the predictions for the samples 5.1a and 5.1c were false, the correct classifications only differed 1 pp.

Table 5.2: Illumination invariant CNN building damage predictions for the images in Figure 5.1. Illustrates the probability of each image belonging to a certain class where the class with the highest probability is chosen as prediction (Underlined = correct classification, green = correctly classified, red = falsely classified).

	No-damage	Minor-damage	Major-damage	Destroyed
(a)	0.03	0.48	0.42	0.07
(b)	0.01	0.18	0.46	0.34
(c)	0.13	0.52	0.35	0.00
(d)	0.19	0.16	0.15	0.50

Then, shifting the attention to Table 5.2, a more successful predictive capacity was demonstrated by the illumination invariant CNN. For these particular images, it classified them all correctly. Furthermore, the probabilities of these predictions appeared to be of an overall higher degree when compared to Table 5.1.

5.1.2 CNN vs G-CNN

Subsequent to the illumination invariant CNN, the G-CNN was compared to a conventional CNN. For this comparison, one image from the test set were chosen, which was then duplicated three times. These duplicate images were then rotated by multiples of 90 degrees. Figure 5.2 shows these images, along with their respective orientation compared to the original image. Both models then made building damage assessment predictions for each image, the output of which is shown in Table 5.3. and Table 5.4.

(a) Original (b) Rotated 90° (c) Rotated 180° (d) Rotated 270°

Figure 5.2: A sample image from the test set, oriented in four different multiples of 90 degree rotations.

As the G-CNN was equivariant to rotations that are multiples of 90 degrees, the prognosis was that the G-CNN would output more stable/equal predictions for each image. This, while the predictions of the CNN were expected to be more inconsistent.

Table 5.3: CNN building damage assessment predictions for the images in Figure 5.2. Illustrates the probability of each image belonging to a certain class where the class with the highest probability is chosen as prediction (Underlined = correct classification, green = correctly classified, red = falsely classified).

	No-damage	Minor-damage	Major-damage	Destroyed
(a)	0.33	0.23	0.24	0.20
(b)	0.50	0.19	0.16	0.15
(c)	0.43	0.23	0.15	0.19
(d)	0.19	0.24	0.33	0.24

Demonstrated by Table 5.3, the conventional CNN assessed the building damage correctly for three out of the four images. Only Fig. 5.2d was falsely classified as majorly damaged instead of not damaged. As expected, the CNN's output was not consistent under rotation of the input image as prediction probabilities varied widely.

Table 5.4: G-CNN building damage assessment predictions for the images in Figure 5.2. Illustrates the probability of each image belonging to a certain class where the class with the highest probability is chosen as prediction (Underlined = correct classification, green = correctly classified, red = falsely classified).

	No-damage	Minor-damage	Major-damage	Destroyed
(a)	0.62	0.25	0.11	0.02
(b)	0.62	0.25	0.11	0.02
(c)	0.62	0.25	0.11	0.02
(d)	0.62	0.25	0.11	0.02

Results for the G-CNN, shown in Table 5.3, were as suspected. As the G-CNN was equivariant to rotations of 90 degrees, its prediction probabilities remained consistent for all images. In case of classifying the images in Figure 5.2, this resulted in all four building damage assessment predictions being correct.

5.2 Model performance

To determine whether our implementation of the illumination invariant G-CNN yielded any improvements over existing methods, its performance was compared to the three other model configurations (see Section 5). For this comparison, all models were trained on varying amounts of training data, after which their performance was measured for every one of these training sizes. To make the comparison as fair as possible, the training, validation and test set were equal for all models and their given training size. Additionally, the training, validation and test set all remained to have equal observations for each class.

5.2.1 Accuracy

Since equal observations were used for each class, the accuracy of each model on the test set could provide an acceptable measure of performance. Figure 5.3 illustrates these accuracies, given different amounts of training sizes.

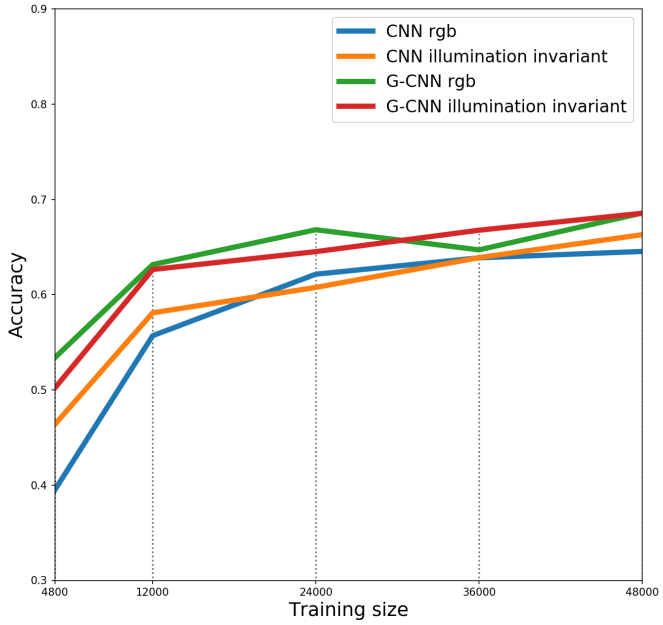


Figure 5.3: All model's accuracy scores, given different training sizes.

As shown in Figure 5.3, when it comes to accuracy, both implementations of the G-CNN consistently outperformed those of the CNN. This was true for all training sizes. Furthermore, for training sizes of 4,800 and 12,000 images, the inclusion of the illumination invariant color space to the CNN provided close to a 7 and 2.5 pp increase in accuracy respectively compared to a conventional CNN. However, the opposite seemed to be true for the G-CNN, where the lack of an illumination invariant color space improved accuracy for all training sizes up to 24,000. For training sizes from 36,000 and upwards, the differences between models seemed to diminish, with all models achieving accuracies more similar to one another.

5.2.2 F1-score

Whereas accuracy provides a valid measure of performance for balanced datasets, its concept is flawed for unbalanced ones. I.e. statistical bias towards classes with a more significant amount of occurrences will skew results. To resolve this, another widely used performance measure is the F1-score, which does not exhibit this drawback. Therefore, in order to enable comparisons with models assessed by the F1-score, this was also measured. These results can be seen in Figure 5.4.

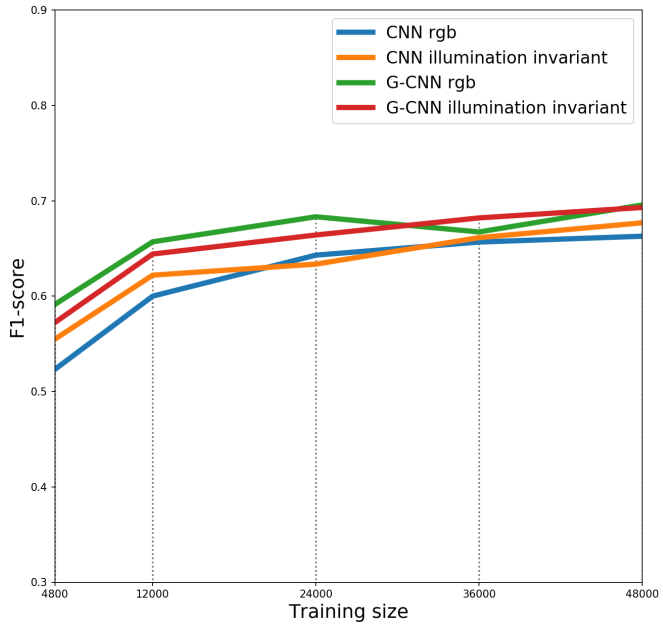


Figure 5.4: All model's F1-scores, given different training sizes.

As class imbalances in the training, validation and test set were resolved in our case, the F1-scores illustrated in Figure 5.4 follow a similar pattern to that of the accuracy's in Figure 5.3: The two G-CNN configurations achieve a higher F1-scores for all training sizes when compared to the conventional CNN. However, the inclusion of the illumination invariant color space does not seem to increase performance in an equally consistent manner.

6 Discussion & Future work

The results show that the p4m convolution layers in the G-CNN can be used to improve a CNN's performance, trained on a smaller dataset. Their incorporation into the model consistently enhances performance, with noticeable improvements for larger training sizes as well.

The addition of the illumination invariant color space to the CNN however did not convincingly improve results. To explain these results, a few of its limitations are discussed. As can be seen in Figure 4.2 and 5.1, some of the illumination invariant color space images exhibit some black spots. Essentially, these black spots are a loss of detail. They are the result of Equation 2.2 exceeding the value of 255, which is the threshold value for any 8 bit image. Some images exhibit these black spots to a more significant degree than ones shown in Figure 4.2 and 5.1. It should be noted that these spots rarely seemed to affect the buildings themselves. Presumably this is because the color values that cause Equation 2.2 to exceed the limit of an 8 bit image are not that often found in buildings. However, any loss of detail is undesired and should be attempted to get fixed. Attempts to resolve the issue by scaling the image were made. However, often this would significantly darken the overall picture, ultimately ruining the illumination invariant aspect of the image. Therefore, with no definitive remedy as of yet, implementing a solution will be left for future work.

In addition to resolving the issues associated with the illumination invariant color space, future work could be focused on improving a CNN's performance by incorporating more advanced data augmentation methods. Some obvious data augmentation methods that were left unused in this project were flip transformations and translations. However, other less explicit data augmentations methods such as Generative Adversarial Networks (GANs) could be used as well. Other improvements could consist of tweaks to the G-CNN architecture. Additionally, more recent works such as that of Berkay Kicanaoglu *et al* [25] have implemented Gauge equivariant Convolutional Networks, which may improve performance even more. Thus, there is still room for further improvement. The intent is that the current work laid a foundation upon which can be build.

7 Conclusion

It has been proven that using p4m G-convolution layers, instead of regular convolution layers, in a CNN can improve its performance when trained on small datasets. By exploiting symmetries, the G-CNN learns features that are equivariant to transformations such as translations, rotations and flips. Hereby, the G-CNN requires less training data, ultimately improving the model's predictive capability. The inclusion of the proposed illumination invariant color space did not demonstrate a similar convincing uplift in performance. Its incorporation showed minor improvements when paired with a conventional CNN, yet minor deterioration in performance when paired with the the G-CNN. Therefore, we can conclude that using the p4m G-convolution layers in a CNN can improve its classification performance for small datasets. Moreover, the benefits a G-CNN can offer over a conventional CNN are not merely limited to satellite imagery. Its use case can also be extended to other image detection or image classification applications where a network's robustness against rotations and flips is essential (e.g. tumor detection in medical scans, marine organism detection, texture classification, etc.).

Bibliography

[1] Denis F. Wolf Patrick Y. Shinzato, Diego Gomes. Road estimation with sparse 3d points from stereo data. https://ieeexplore.ieee.org/document/6957936, 2014.

[2] Niko Sunderhauf Sareh Shirazi Edward Pepperell Ben Upcroft Chunhua Shen Guosheng Lin Fayao Liu Cesar Cadena Ian Reid Michael Milford, Stephanie Lowry. Sequence searching with deep-learnt depth for condition- and viewpoint invariant route-based place recognition. https://www.cv-foundation.org/openaccess/content_cvpr_workshops_2015/W11/papers/Milford_Sequence_Searching_With_2015_CVPR_paper.pdf.

[3] L. Magdalena M.A. Sotelo, F.J. Rodriguez. Virtuous: vision-based road transportation for unmanned operation on urban-like scenarios. https://ieeexplore.ieee.org/document/1303538, 2004.

[4] Jianwei Zhang Calin Rotaru, Thorsten Graf. Color image segmentation in hsi space for automotive applications. https://link.springer.com/article/10.1007/s11554-008-0078-9, 2008.

[5] Sung-Eui Yoon Taeyoung Kim, Yu-Wing Tai. Pca based computation of illumination-invariant space for road detection. https://ieeexplore-ieee-org.proxy.uba.uva.nl:2443/document/7926659, 2017.

[6] Antonio M. Ĺopez José M Álvarez Alvarez. Illuminant-invariant model-based road segmentation. https://ieeexplore.ieee.org/document/4621283, 2010.

[7] R. Baldrich J.M. Alvarez, A. Lopez. Road detection based on illuminant invariance. https://ieeexplore.ieee.org/document/5594640, 2008.

[8] D.G. Lowe. Object recognition from local scale-invariant features. https://arxiv.org/abs/1902.08802, 2019.

[9] B. Triggs N. Dalal. Histograms of oriented gradients for human detection. https://ieeexplore.ieee.org/document/1467360, 2005.

[10] Shaohui Mei Shun Zhang Yifan Zhang Zhi Zhang, Ruoqiao Jiang. Rotation-invariant feature learning for object detection in vhr optical remote sensing images by double-net. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8936929, 2019.

[11] Stefan Roth Uwe Schmidt. Learning rotation-aware features: From invariant priors to equivariant descriptors. https://ieeexplore.ieee.org/document/6247909, 2012.

[12] Jyri J. KivinenChristopher K. I. Williams. Transformation equivariant boltzmann machines. https://link.springer.com/chapter/10.1007/978-3-642-21735-7_1, 2011.

[13] Skibbe. , 2013.

[14] Gens & Domingos. , 2014.

[15] Colin McManus Ben Upcrof Winston Churchill Paul Newman Will Maddern, Alexander D. Stewart. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. http://www.robots.ox.ac.uk/~mobile/Papers/2014ICRA_maddern.pdf.

[16] D. Hodgkiss S.T. Henderson. The spectral energy distribution of daylight. https://iopscience.iop.org/article/10.1088/0508-3443/15/8/310, 1964.

[17] Javier Romero Javier Hernández-Andrés and Jr Juan L. Nieves, Raymond L. Lee. Color and spectral analysis of daylight in southern europe. https://www.researchgate.net/publication/11945067_Color_and_spectral_analysis_of_daylight_in_southern_Europe, 2001.

[18] Steve Collins Sivalogeswaran Ratnasingam. Study of the photodetector characteristics of a camera for color constancy in natural scenes. https://www.osapublishing.org/josaa/abstract.cfm?uri=josaa-27-2-286, 2009.

[19] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. https://arxiv.org/pdf/1602.07576.pdf, 2016.

[20] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant (cnns) for digital pathology. https://arxiv.org/abs/1806.03962.pdf, 2018.

[21] Ritwik Gupta et al. xbd: A dataset for assessing building damage from satellite imagery. https://arxiv.org/pdf/1911.09296.pdf, 2019.

[22] Worldview-3. https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/105/DigitalGlobe_Spectral_Response_1.pdf.

[23] Laurens van der Maaten Kilian Q. Weinberger Gao Huang, Zhuang Liu. Densely connected convolutional networks. https://arxiv.org/abs/1608.06993, 2016.

[24] Claude E. Duchon. Lanczos filtering in one and two dimensions. https://www.researchgate.net/publication/252898828_Lanczos_Filtering_in_One_and_Two_Dimensions/link/00b4953c699969aae5000000/download, 1979.

[25] Berkay Kicanaoglu Max Welling Taco S. Cohen, Maurice Weiler. Gauge equivariant convolutional networks and the icosahedral cnn. https://arxiv.org/pdf/1902.04615.pdf, 2019.

[26] B. S. Veeling. Group equivariant convolutional neural networks for keras: keras gcnn. https://github.com/basveeling/keras-gcnn, 2018.

Appendices

A Preprocessing pipeline for validation and test set

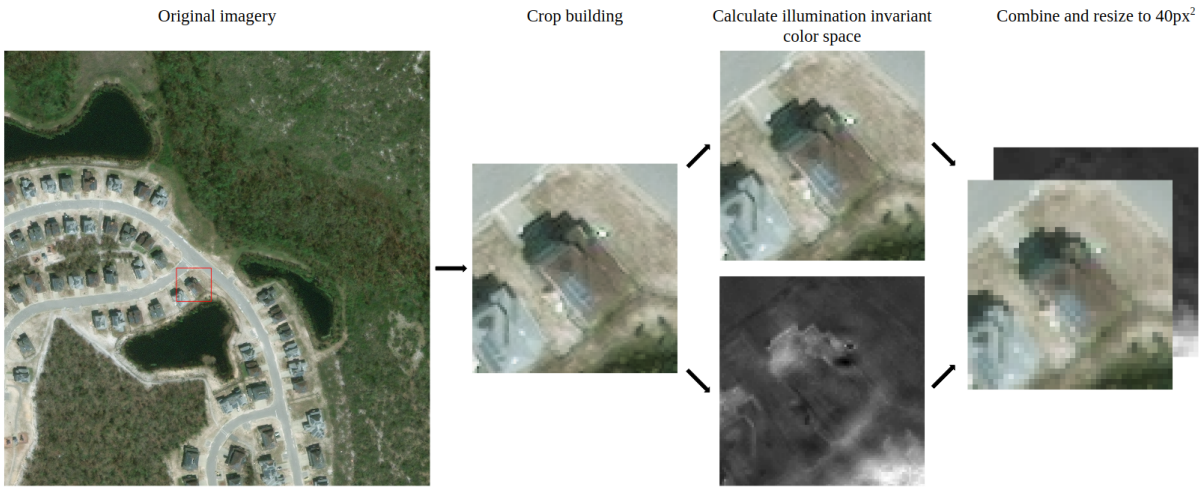


Figure 7.1: The preprocessing pipeline for extracting buildings from satellite imagery without data augmentation. This process was used for all building images that made up the validation and test set.

B Software used

- Miniconda
- Python 3.6.10
- Keras 2.1.6
- Keras-gcnn 1.0 [26]
- Tensorflow 1.10
- Tensorflow-gpu 1.10