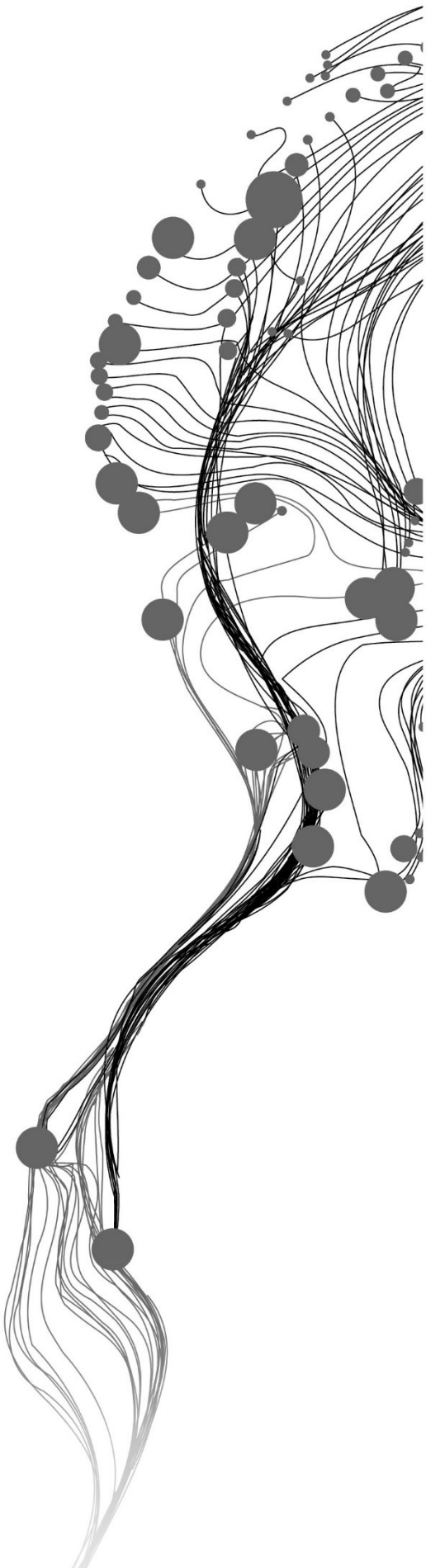
DETECTION OF INFORMAL SETTLEMENTS FROM VHR SATELLITE IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

NICHOLUS ODHIAMBO MBOGA
March, 2017

SUPERVISORS:
Dr. C. Persello
Prof. Dr. Ir. A. Stein
ADVISOR:
J.R. Bergado MSc



DETECTION OF INFORMAL SETTLEMENTS FROM VHR SATELLITE IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

NICHOLUS ODHIAMBO MBOGA
Enschede, The Netherlands, March, 2017

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geo-Informatics

SUPERVISORS:
Dr. C. Persello
Prof. Dr. Ir. A. Stein
Advisor: J.R. Bergado MSc

THESIS ASSESSMENT BOARD:
Prof.Dr.Ir. M.G. Vosselman (Chair)
Ms M. Kuffer MSc (External Examiner, University of Twente, ITC-PGM)

ABSTRACT

Convolutional neural networks (CNNs), widely studied in the domain of computer vision, are more recently finding application in the analysis of high resolution aerial and satellite imagery. In this research, we investigate a deep feature learning approach based on CNNs for the detection of informal settlements in Dar es Salaam, Tanzania. Informal settlements represent areas whose quality of life and housing is mostly below acceptable standards. Thus, information about their location and extent helps in decision making and planning process for their upgrading. Distinguishing the different urban structure types is challenging because of the abstract semantic definition of the classes as opposed to the separation of standard land-cover classes. This task requires the extraction of complex spatial-contextual features (or underlying representations in an image), which can be done through hand-crafting (hand-engineering) and feature learning. Whereas hand-crafting is a laborious process that requires testing of many parameters values with a trial and error approach, feature learning allows the automatic detection of such representations from the data. CNNs allow us to automate the extraction of spatial-contextual features. Moreover, they have shown the capability to learn highly informative features resulting in excellent performance, often outperforming techniques based on hand-engineered features. To this aim, we first designed the architecture of the CNN, optimized its hyper-parameters and trained it in an end-to-end fashion to detect informal settlements in VHR images. The obtained results were compared against state-of-the-art methods (i.e. support vector machines (SVMs) with radial basis function (RBF) kernel) relying on hand-crafted features. The experimental results show that SVM relying on Grey Level Co-Occurrence Matrix (GLCM) features results in high classification accuracy. However, CNN outperforms this approach especially when a higher number of convolutional layers and a large training set was used. The highest overall accuracy obtained by SVM relying on GLCM is 86.65% while CNN results in 91.71%. A deeper network allows the CNN to learn a hierarchy of spatial contextual features to allow for better discrimination of classes with a high level of semantic abstraction, while an adequate training set allows for the optimal determination of the parameters of the network. We conclude that CNNs, trained in an end-to-end fashion, are able to effectively learn the spatial-contextual features for accurate discrimination of informal settlements from other settlement types in VHR images.

Key words-Image classification, informal settlements, convolutional neural networks, deep learning, high resolution satellite imagery.

ACKNOWLEDGEMENTS

Glory to God for the well-being, and to finally write this Thesis. I wish to appreciate Netherlands Fellowship Program (NUFFIC) for providing financial support towards my studies in the Netherlands. Sincere appreciation to my supervisors Dr Persello and Prof. Stein, and my advisor John Ray, for your mentorship. The desire to master machine learning in the domain of Geospatial engineering, will grow strong with each passing day. To say the least, I have grown immeasurably as a scholar and an engineer, and will treasure the advice proffered.

I would also like to thank Ms Monika Kuffer and Dr Richard Sliuzas for providing the Quickbird dataset. I appreciate the esteemed staff at Faculty of Geo-Information Science and Earth Observation, ITC-University of Twente, for they played a part in one way or another, during my studies. In addition, I salute my classmates from all over the world, with whom we toiled daily. I appreciate the jovial spirit and camaraderie of the Kenyan community at ITC, which maintained wonderful cheer. Lastly, I extend warm appreciation to dear family for always being there and providing moral support, and I dedicate this to you. Thank you.

"Dare to dream"-Anonymous

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Motivation and Problem statement

Escalating urbanisation has resulted in growth of informal settlements in developing countries. The lack of spatial information on informal settlements has created the need for techniques that provide this information in an accurate and timely way.

An informal settlement can be defined as "a contiguous settlement where the inhabitants are characterised as having inadequate housing and basic services. A slum is often not recognised and addressed by the public authorities as an integral or equal part of the city" (UN-HABITAT, 2003). Besides, slums represent the most underprivileged examples of informal settlements. Some of the unacceptable conditions present are poor access to safe water, sanitation and infrastructure. Moreover, the structural quality of housing is sub-standard, overcrowding and uncertain residential status are common characteristics (UN-HABITAT, 2012). Slums are linked to poverty (Kohli, Sliuzas, Kerle, & Stein, 2012), and information about their location and extent aids in planning and decision making for their upgrading (Hofmann, Strobl, Blaschke, & Kux, 2008).

Availability of Very High Resolution (VHR) satellite imagery (Lu, Hetrick, & Moran, 2010) provides the opportunity to distinguish slums from formal settlements based on physical (morphological) characteristics of the urban structure. Slums are mostly characterised by small and clustered buildings with an irregular spatial pattern and almost no presence of vegetation. This is different from formal areas where the buildings are large, there is presence of vegetation, and the spatial pattern is regular (Gueguen, 2015; Kuffer & Barros, 2011). In high spatial resolution images, a pixel is mostly smaller than the object of interest, and contains little contextual information to accurately distinguish such a class (Vatsavai, Bhaduri, & Graesser, 2013). Furthermore, there is a high intra-class variance and low inter-class variance (Lu et al., 2010; Tokarczyk, Wegner, Walk, & Schindler, 2013). Consequently, extraction of spatial-contextual features is necessary to improve the classification process (Bergado, Persello, & Gevaert, 2016; Shekhar, 2012) from VHR satellite imagery. Spatial information refers to the spatial arrangement of the spectral information in a scene while the contextual information describes the information that is extracted from a neighbourhood (Haralick, Shanmugan, & Dinstein, 1973).

Spatial-contextual features can be generated through hand-crafting (hand-engineering) and feature learning. Features are underlying representation in data that facilitate the classification task. While hand-crafting is a laborious process that requires the values of the parameters to be determined manually through trial and error, feature learning enables them to be automatically detected from the input data (LeCun, Bengio, & Hinton, 2015). Machine learning methods are able to automatically detect patterns in data, and make use of the discovered patterns for the classification task. An example is deep feature learning methods, which learn a hierarchy of features by automatically constructing high-level features from low-level ones (Castelluccio, Poggi, Sansone, & Verdoliva, 2015; Ji, Xu, Yang, & Yu, 2013). They are based on artificial neural networks, which have the advantage of classifying multi-source data because they are non-parametric, nonlinear and perform well in domain-adaptation problems (Vatsavai et al., 2011).

Detection of informal settlements can be considered as a land use classification problem because it requires the definition of classes with a higher level of semantic abstraction. A land use class mostly contains several types of different land cover types, covering different extents (scale), having different orientations. Thus, unlike land cover it is a challenging process to infer the class label of a pixel by relying only on the spectral signature. Therefore, better features are needed to enable the discrimination of such complex classes

present in a given scene (Castelluccio et al., 2015). Figure 1.1 shows examples of slums and formal settlements respectively, and illustrates the presence of different land cover types in each scene.

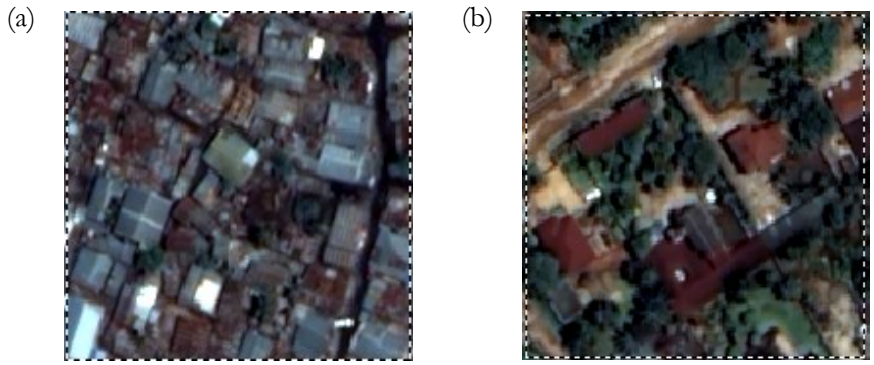


Figure 1.1: 100 × 100 m scenes of a (a) slum and (b) formal settlement

Convolutional neural networks (CNNs) are able to automate the extraction of spatial-contextual features by learning a hierarchy of simple to complex features from the raw input images (Ji et al., 2013). They have been successfully applied in fields of computer vision, speech recognition and discovery of drugs (Deng, 2014; Schmidhuber, 2015). However, use of such deep learning approaches needs to be investigated in the detection of informal settlements in an urban scene using VHR satellite imagery.

1.2. Research identification

The research focusses on investigating the applicability of deep learning approaches to the problem of detecting informal settlements in an urban scene using VHR satellite images. We use Quickbird VHR image for our experiments, acquired over the city of Dar es Salaam, Tanzania. We develop a methodology for detecting informal settlements based on CNNs. We optimize the network design and experiment with several hyper-parameters of the CNN and carry out a performance comparison with state of the art methods relying on hand-crafted features. Figure 1.2 illustrates manually digitized informal settlements. It is an example of a land use problem whereby the proposed approach should be able to classify pixels as belonging to an informal class or other classes.

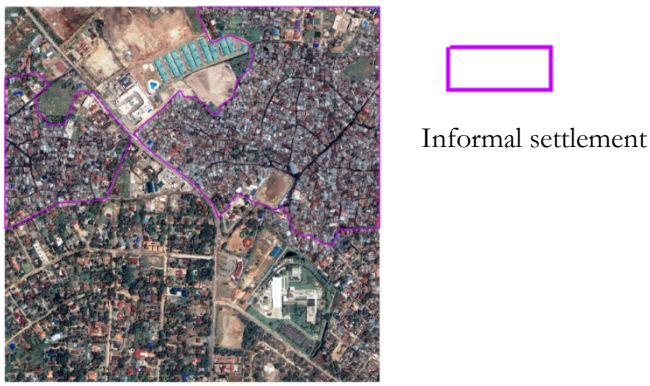


Figure 1.2: A 1200 × 1200 m image tile of Dar es Salaam illustrating a manually digitized informal settlements, QuickBird image: 2007

1.3. Research objectives

The main objective of this study is to investigate deep feature learning methods for the detection of informal settlements from VHR satellite imagery. From this, we derive four specific objectives which are:

i. Review Convolutional neural networks (CNN) and their recent variants.
ii. Develop a methodology for detecting informal settlements from VHR images
iii. Experiment with different CNN architecture designs and hyper-parameters.
iv. Compare the performance of CNN against state of the art methods relying on hand-crafted features.

1.4. Research questions

Referring to the objectives, the following research questions are addressed.

Specific objective 1

i. How have the deep models been applied in the analysis of satellite imagery?
ii. What are the building blocks of a CNN?

Specific objective 2

i. How should the classes be defined?

Specific objective 3

i. What effect does varying the hyper-parameters have on the classification results?
ii. What considerations should be made when designing a new CNN architecture?

Specific objective 4

i. How do the methods compare in terms of accuracy and on previously unseen data?

1.5. Innovation aimed at

This research applied most recent deep feature learning methods for informal settlement detection from VHR satellite imagery. This is indeed novel considering the level of difficulty and challenge that land use classification requires the definition of classes with higher level of semantic abstraction. Deep learning methods have been commonly applied in natural language processing and computer vision domains, but this research applied them for detecting informal settlements. Also, no previous research has used CNNs for detection of informal settlements from VHR images.

1.6. Method adopted

We conducted a literature review of convolutional neural networks (CNNs). This was followed by the design and optimization of hyper-parameters of a CNN, which was trained in an end-to-end fashion. A detailed comparison between the classification results of CNN and support vector machines (SVM) relying on hand-crafted features was done. An overview of the methodology of the study is presented in Figure 1.3.

1.7. Thesis structure

This thesis consists of seven chapters. In Chapter 1, we provide the motivation, research problem, objectives and the research questions. In Chapter 2, we start by introducing the concept of informality in Dar es Salaam, Tanzania followed by a concise review of convolutional neural networks. Chapter 3 describes the data and software used in the execution of the research. Chapter 4 describes the methodology followed to carry out the experiments. Results are presented in Chapter 5 and the discussion in chapter 6. Lastly, conclusions drawn from the study and recommendations for future research opportunities are presented in Chapter 7.

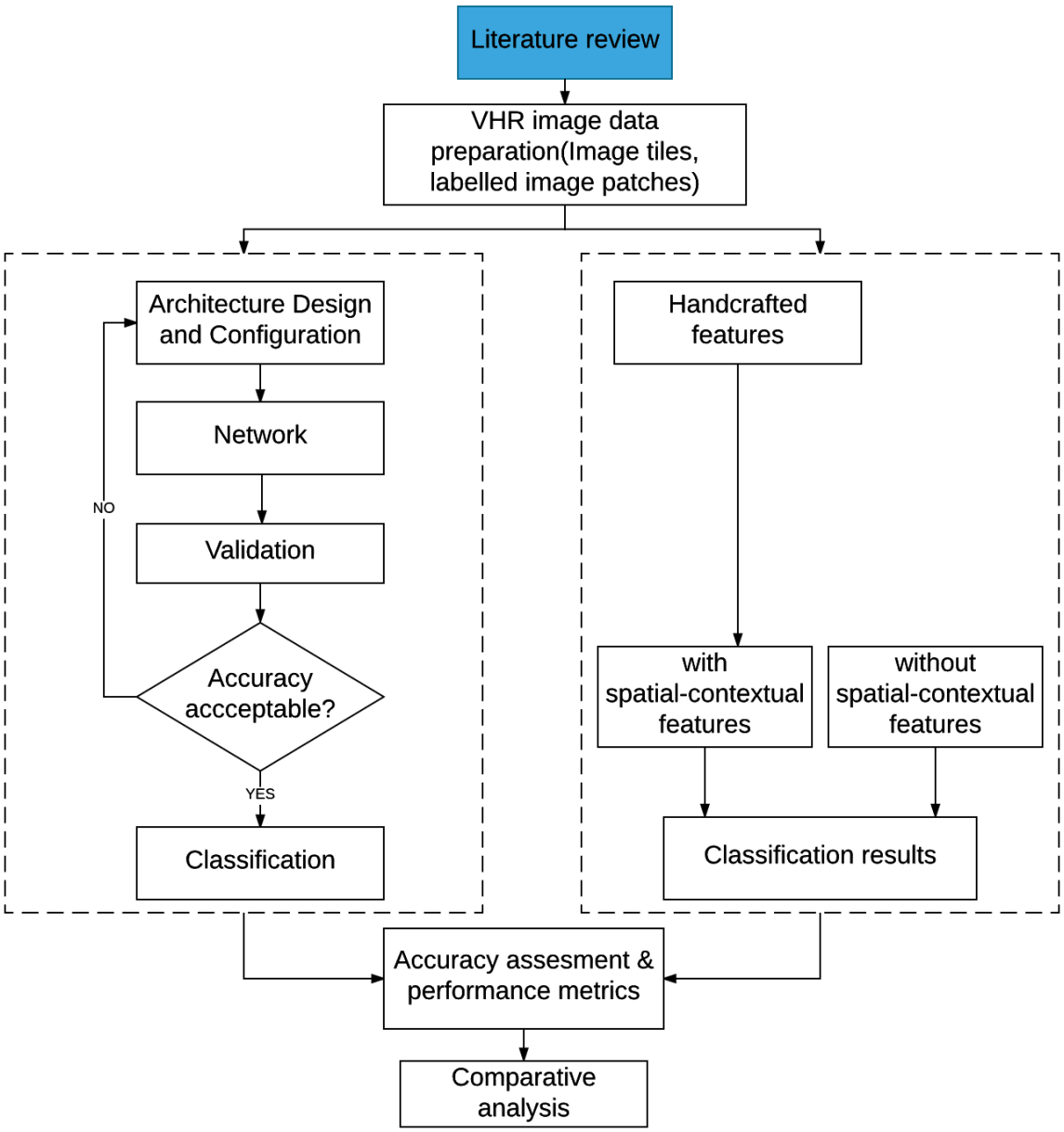


Figure 1.3: Diagram illustrating the general methodology of this study

2. LITERATURE REVIEW

This chapter provides a theoretical background for this research. The concept of informal settlements as described in the urban planning domain is discussed in Section 2.1. Next, a concise review of CNN models, and their application in the domain of remote sensing is provided is provided in Section 2.2.

2.1. Informality in Dar es Salaam, Tanzania

Before the colonial period, land in Sub Saharan Africa was controlled using customary laws. Later, during this colonial period, a system of managing land that was parallel to the existing customary laws was introduced. The formal land law was based on the British example and was introduced to administer land (Sliuzas, 2004). Formal land has security if tenure that is issued by the public authorities, whereas informal land is either administered using native customary law, or bears unclear tenure status (Kironde, 2006). In addition, a land use plan is normally prepared before settlements are raised in formal areas. On the contrary, informal settlements are usually set up first, then later attempts are made to design a land use plan, while makes them unplanned (Sliuzas, 2004). High rates of urbanisation have drawn more people to the urban centres mainly in search of work and a better life. Even though some of the people are able to afford to live in the well planned settlements, the majority lack the financial means to do so. Consequently, they seek affordable shelter which are mostly located in the informal areas and over 80% of the buildings in Dar es Salaam are located in informal areas. Similarly, a high proportion of the city's population lives in unplanned areas, the figure being estimated well over 80 % (Kironde, 2006).

There are diverse terms that are used to refer to the concept of informal settlements in specific parts of the world. They include "squatter settlements", "favelas", "poblaciones", "shacks", "barrios", "bajos", "bidonvilles" and "slums" (UN-Habitat, 2015). A study by Hill & Lindner (2010) considers the term informal and slum to imply the same thing but prefer informal because slum or 'mbanda' is hardly used to describe such settlement types in Tanzania. Elsewhere, in the study by Kuffer, Barros, & Sliuzas (2014), the term unplanned settlements is used. It is construed to imply areas where development of buildings occurs without following a plan, consequently having an irregular layout and inadequate services and infrastructure.

Unplanned settlements in developing cities have a large aerial extent, and in some cases form the major urban land use. Informal settlements grow fast, and can sometimes be scattered within the formal settlement areas. There is a shortage of information regarding these unplanned settlements (Kombe & Kreibich, 2001; Kuffer & Barros, 2011). Mapping informal settlements provides spatial information (i.e. about their location and extent) that is used inform the decision making process of by the local authorities. There is a need to explore the use of geo-information technologies to map the physical state of informal settlements (Sliuzas, 2004), including automatic methods. Several works have been carried out in an attempt to map unplanned settlements from satellite imagery. However, the authors have focused only on the morphological characteristics in their attempt to define such settlements. The legal dimension that is attached to the definition of unplanned settlements is often ignored because it cannot be directly derived from the satellite image (Kuffer et al., 2014).

Unplanned settlements have specific characteristics depending on the geographical area. However, they tend to exhibit some similarities (Hofmann, 2014). Morphological characteristics that generally distinguish between planned and unplanned settlements are displayed in Table 2.1. These physical characteristics are also discussed in (Kombe & Kreibich, 2001).

Table 2.1: Morphological characteristics of unplanned and planned settlements, adapted from (Kuffer & Barros, 2011)

Residential Type	Spatial characteristics in VHR images
Unplanned areas	• High densities (roof coverage densities at least 80% and more) • Organic layout structure (no orderly road arrangement non-compliance with set-back standards) • Lack of public (green) spaces in the vicinity of the residential areas • Small sub-standard building sizes
Planned areas	• Low moderate density areas • Regular layout pattern (showing planned regular roads and compliance with setback rules) • Provision of public (green) spaces within or in vicinity of residential areas • Generally larger building sizes

It is evident that unplanned settlements form part of the urban residential land use. In addition, it is a fact that there are conflicting definitions of what constitutes an unplanned settlement, and this is also dependent on the locality. However, this research intends to contribute to the first step of detecting unplanned settlements making use of VHR. The terms informal settlement and formal settlements will be used, bearing in mind that only their morphological characteristics can be directly inferred from the satellite imagery. The available land use reference dataset also uses these terms to define the classes. As an exception, the use of the term slum shall be construed to imply an informal settlement. The legal definition as to what constitutes an unplanned settlement will be considered as being beyond the scope of this research.

2.2. A review of convolutional neural networks

2.2.1. Background

CNNs are artificial neural networks that draw inspiration from the biological neuron, and represent information using several hierarchical layers. A typical biological neuron is made of a cell body, a tubular axon and dendrites. Figure 2.1 shows a generalized diagram of an artificial neuron that tries to relate in a simplified way the relation between a biological neuron and an artificial neuron. The artificial neuron is the foundation of the CNN.

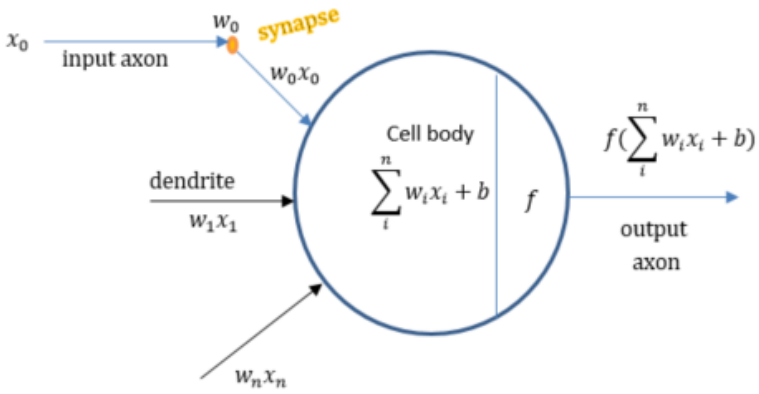


Figure 2.1: A generalized diagram of an artificial neuron, adapted from (CS213n, 2016)

In principle, the inputs to the neuron are represented as $x_0, x_1, \ldots, x_n$. The strength of the connection at the synapse is given as $w_0, w_1, \ldots, w_n$, which denotes the weights whereas b denotes the bias term. A summing operation is applied to the inputs (although a product operation can be applied instead) which results in a linear output. A nonlinear function (activation), f is applied to the output, resulting in a nonlinear transformation. Unsaturated nonlinear activations are preferred over saturated nonlinear transformations because they do not suffer from the vanishing gradient problem. The Rectified Linear unit (RELU), given as $g(z) = \max\{0, z\}$, is useful in optimizing models that are gradient-based because they remain almost linear. Faster training of networks is observed when RELU nonlinearity is used as compared to hyperbolic tangent units (Krizhevsky, Sutskever, & Hinton, 2012). Examples of saturating nonlinearity, namely the hyperbolic tangent and the sigmoidal function, are given in Equation 2.1 and Equation 2.2 respectively.

$$f(x) = \tanh(x) \qquad \text{Equation 2.1}$$

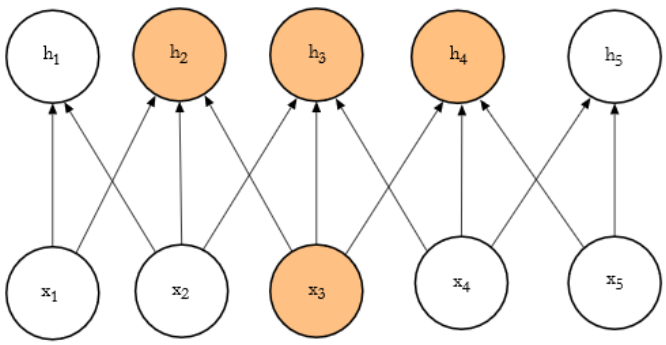
$$f(x) = (1 + e^{-x}) \qquad \text{Equation 2.2}$$

Biological terminology and the artificial neural network terminology are presented in Table 2.2.

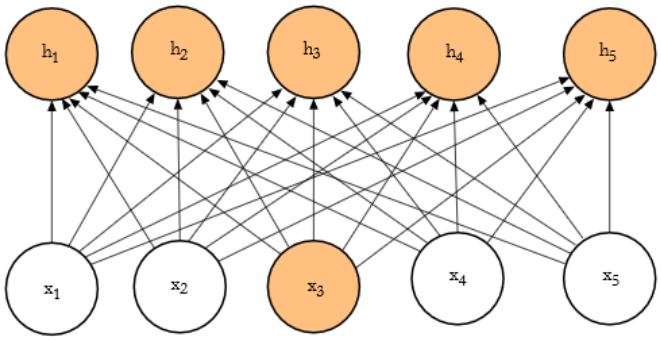
Table 2.2: Similar Biological and Artificial neural network terminology. Adapted from (Mehrotra, Mohan, & Ranka, 1997)

Biological terminology	Artificial neural network terminology
• **Neuron**	Node/unit/cell/
• **Synapse**	Connection/edge/link
• **Synaptic efficiency**	Connection strength/weight
• **Firing Frequency**	Node output

In CNNs, at least one of the layers uses convolutions rather than matrix multiplication. As a result, CNNs are characterised by three desirable properties namely sparse interaction, parameter sharing and equivariant representations. During a convolution operation, the use of a kernel with a smaller dimension than the input reduces the number of connections, and hence the number of parameters when determining the output. This results in sparse connectivity, as shown in Figure 2.2 (a). Using a kernel size of three implies that the input is connected to only three units. However, in a fully-connected layer, a unit is connected to all the units in the subsequent layer, as shown in Figure 2.2 (b), a unit is fully connected to the units in the next layer, thus lacking sparse connectivity (Bengio, Goodfellow, & Courville, 2015).



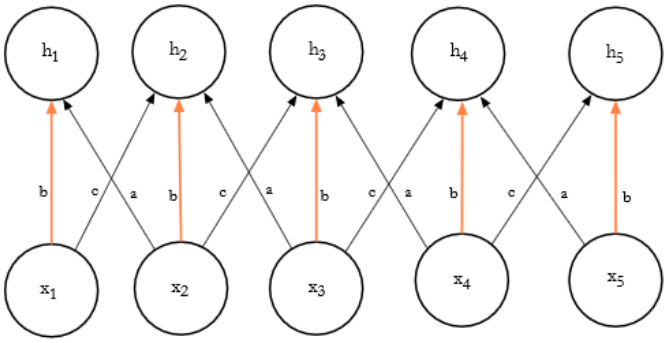
(a)



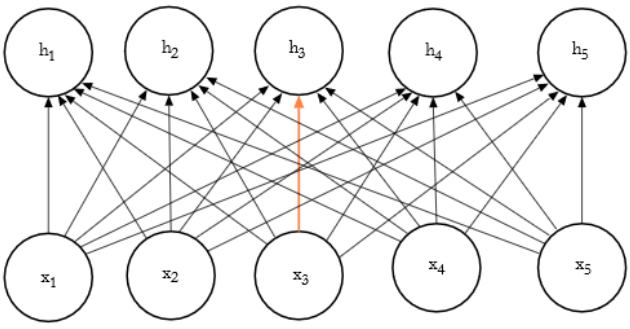
(b)

Figure 2.2: Sparse connectivity. An example of a convolution operation using a kernel size of three is used shown in (a), while in (b), fully-connected units are shown whereby a matrix multiplication is carried out, adapted from (Bengio et al., 2015).

For parameter sharing, the same set of weights is learned for each location in the input image in a particular layer. Figure 2.3 (a) illustrates the network that has a convolution where the kernel is of size three. The parameters learned, a, b and c are applied at every location of the input layer. The idea is that if a feature occurs in one particular location in the image, for example, then it is likely to occur in another part of the image (Bengio et al., 2015). On the other hand, the fully-connected model, shown in Figure 2.3 (b) does not have parameter sharing. Each parameter is used only once. The orange line shows where a parameter is used in more than one occasion.



(a)



(b)

Figure 2.3: An illustration of parameter sharing which is present in the convolutional network (a) but absent in the fully connected network (b), adapted from (Bengio et al., 2015).

The equivariance property of a convolution enables the detection of features when they occur at different locations (Bengio et al., 2015). This implies that when there is a transformation on the input image such as a shift before applying a desired function, then there should be a corresponding predictable transformation in the output after applying the said function. Translational equivariance is a property of CNNs introduced through pooling (Kivinen & Williams, 2011).

During pooling, the summary statistics of the adjacent outputs are used to determine the activations to be propagated to the next layer. This can be done through max-pooling and average pooling. A max-pooling over an input region of size $p \times p$ is whereby the most dominant signal is propagated to the next layer is carried out. On the other hand, average pooling returns the average of the signals in the window being considered. When pooling is carried out using a stride s, where $s > 1$, it results in down-sampling of the input with a factor s. This reduces the spatial dimension of the output. Loss of spatial information might affect tasks that require precise localisation such as semantic segmentation (Pinheiro & Collobert, 2014).

Supervised training of a CNN consists of a forward propagation and backward propagation. During forward propagation, the network takes in a set of input x and produces a set of output y. When the inputs go through the network, a scalar cost is produced. Backpropagation allows the information to flow backwards into the network for the computation of gradients. The optimization algorithm that is used for learning by the CNN is the stochastic gradient descent (SGD). Parameters associated with SGD are the learning rate ϵ, which affects how fast the learning takes place, and momentum α used to accelerate the learning. The learning rate is decayed linearly at every iteration, τ because SGD introduces random noise. A cross-entropy between the training data and the model's predictions is used as the objective function. The gradients of the cost function with respect to the parameters should be large enough to guide the learning algorithm (Bengio et al., 2015). Detailed formulations are described in Section 4.1.1.

When training a large network with few samples, there is a risk of overfitting. This means that it loses its generalization ability. Several techniques are used to mitigate overfitting. One of them is data augmentation. It involves expanding the variety of the training set, such that for each high dimensional input feature, $\boldsymbol{x}$, which has a corresponding label of y, a transformation is applied on $\boldsymbol{x}$, such that new$(\boldsymbol{x}, y)$ pairs are generated (Bengio et al., 2015). Examples of data augmentation include random sampling, random transformation and noise infection (Volpi & Tuia, 2016) and performing principle component analysis (PCA) on the images, and adding the multiple PCA available (Krizhevsky et al., 2012). Dropout helps to mitigate co-adaptation of neurons that results in interdependent filters in the same layer (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). This involves turning off a given percentage of neurons and their connections by setting the parameter to a value in the range$[0,1]$. Consequently, a model that is less correlated is trained at every epoch. A neuron learns a set of useful features with a random set of other neurons. A value of 0.5 is used in (Krizhevsky et al., 2012). Another way is by use of early stopping. This is whereby during training, the algorithm is run until the error on the validation set does not improve for a given number of epochs, e_n. Thirdly, L2 parameter norm penalty penalizes the high values of parameters towards zero.

A CNN comprises several layers arranged hierarchically, whereby the lower convolutional layers describe low level features such as edges while the higher convolutional layers learn a set of abstract/complex features (LeCun et al., 2015). The receptive field refers to the area in the previous layer that is connected to a neuron in the subsequent layer. Neurons in a CNN have a local receptive field instead of a global receptive field. In CNNs where there is sub-sampling, it is intuitive that, higher layers have a larger receptive field than the lower layers (Long, Shelhamer, & Darrell, 2015). CNN have gained attention because they are able to learn invariant features that are useful across an input image. It is also possible to control the capacity of the network by varying a set of hyper-parameters that determine its depth such as the number of layers, kernel dimensions and the size of the input image, which has an impact on the classification accuracy of the CNN. The representations obtained by a CNN are learned through a hierarchy

of convolutional filters from the input image. Weights are learned in an end to end fashion minimizing the loss function of the model.

We present concisely, in Section 2.2.2, examples of patch-based and image-based CNNs, where we highlight some applications in satellite imagery analysis. In patch-based (patch-wise) CNN, a fixed area of an image (patch) is used as an input during training. At inference time, a label is assigned to the central pixel of the patch. On the other hand, an image of an arbitrary size can be used during training in image-based (image-wise) CNN, producing an output that has corresponding dimensions as the input (Long et al., 2015; Sherrah, 2016).

2.2.2. A brief overview of CNN applications.

The CNN implemented in (Krizhevsky et al., 2012), was used to successfully classify images the ImageNet LSVRC-2010 contest. This is one of the factors that encouraged research into the use of CNN for classification of images. Some research has been carried out in the domain of computer vision involving CNN. Similar to this architecture is the model by Bergado et al. (2016) which is developed for land cover classification of high resolution aerial images. Images from the ISPRS_VAIHINGEN Benchmark dataset are used to train and test the model. These two models have one instance of CNN that is composed of a series of convolutional layers followed by a fully connected layer and finally, an n-way softmax classifier. Furthermore, they are patch-based. The CNN part extracts the hierarchy of features while the fully connected layers learn the classification rule with respect to the features learnt. Hold-out cross validation is used in optimizing the hyper-parameters and regularization parameters. Another instance where CNN is used for land cover classification is in (Paisitkriangkrai, Sherrah, Janney, & Hengel, 2016). CNN have also been used for land use classification tasks, for example (Castelluccio et al., 2015; Luus, Salmon, Van Den Bergh, & Maharaj, 2015).

Recent CNN variants include fully convolutional networks (Long et al., 2015; Sherrah, 2016), deconvolutional neural networks (Noh, Hong, & Han, 2015; Volpi & Tuia, 2016; Zeiler, Taylor, & Fergus, 2011) and recurrent convolutional neural networks (Pinheiro & Collobert, 2014). The DCN and the FCN carry out image-wise training and inference instead of using patches. Although the performance of deep learning algorithms is high in image classification, there is no adequate research on their application in VHR images (Hu, Xia, Hu, & Zhang, 2015) and their suitability for complex urban scenes. This research uses deep convolutional neural network to detect informal settlements from very high resolution satellite imagery.

3. DATA AND SOFTWARE

In this chapter, a description of raw data, ground reference data, software and deep learning framework used is provided.

3.1. Data description

We used Quickbird satellite image of Dar es Salaam, Tanzania, acquired in 2007. The multispectral image has four bands: Blue, Green, Red and Near Infrared. The image is pan-sharpened and has a spatial resolution of 0.60 m. Labelled reference information was obtained using both visual interpretation and a land use reference map (Sliuzas, 2004; Sliuzas, Hill, Lindner, & Greiving, 2016). We consider 3 tiles of 2000 ×2000 pixels. Each tile covers an area on the ground of 1.2 × 1.2 km. Four classes are available from the reference map, namely "formal settlement", "informal settlement", "other urban" and "vacant/agriculture. A summary of the dataset is presented in Table 3.1.

Table 3.1: Description of the dataset used in the study

Dataset	Description	Status	Year	Location
Quick-Bird	0.60 m resolution, 4 bands{B,G,R,NIR}	Available	2007	Dar es Salaam, Tanzania
Land Use	Vector	Available	2002	Dar es Salaam, Tanzania

In the preliminary set of experiments, only the classes "formal" and "informal" are considered. The other two classes are not considered because they cannot be accurately discriminated only on the basis of physical features derived from remotely sensed images. In the second set of experiments, the classes "formal settlement", "other urban" and "vacant/agriculture" are merged into one class. We evaluate the ability of the classifier to distinguish informal settlement class from all the other urban classes. The input data is normalized in the range [0, 1]. Stratified sampling is used to generate training samples from the dataset. To evaluate the accuracy, we carry out a full image test on each of the tiles.

3.2. Software

The deep learning framework is based on Theano and Keras library. Python language is mainly used for programming. R- Software (version 3.2.3) is used to extract the Grey Level Co-occurrence Matrix (GLCM) features. Graphical plots are prepared using Matlab R2014b. ArcGIS 10.4.1 is used to prepare the land use reference data. In Figure 3.1, we show the raw images and their corresponding reference dataset.

Tile	Raw image	Ground reference data
1		
2		
3		
	Informal formal Other_Urban/ Vacant	

Figure 3.1: The raw images and the corresponding ground reference data

4. METHODOLOGY

This chapter describes the set of experiments carried out towards the main objective of detecting informal settlements from VHR satellite images. Preliminary experiments are carried out which influence design of the final network. Experiments are conducted using the designed CNN and performance is compared to SVM relying on handcrafted features (i.e. GLCM).

4.1. Preliminary experiments: informal settlements vs formal settlements

4.1.1. CNN hyper-parameter optimization

We build our CNN using the architecture from (Bergado et al., 2016) as a foundation. We use patch based classification approach. Figure 4.1 illustrates the general architecture of the adopted CNN. The input data consists of a 3-dimensional array of size $(m \times m \times b)$, where b is the number of bands and m is the width and height of the input patch. The first convolutional layer comprises of k filters of dimensions $(f \times f)$. The first convolutional layer performs a convolution over the 3D input volume.

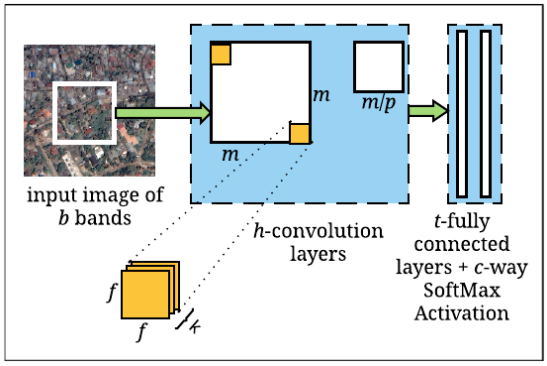


Figure 4.1: Diagram illustrating the adopted CNN

A nonlinear activation function, the Rectified Linear Unit (RELU), is applied on the resulting linear activations. A max-pooling over an input region of size $p \times p$ whereby the most dominant signal is propagated to the next layer is carried out. Pooling with a stride of s, where $s > 1$ results in down-sampling with a factor s. This is repeated in the subsequent convolutional layers. The output of the final convolutional layer is flattened to a one dimensional vector containing the extracted features and fed into t fully connected layer with z filters.

The output of the last fully connected layer are normalized using a soft-max activation function. It has c units, representing number of classes. It returns the posterior probability of the classes and is expressed in Equation 4.1 as:

$$p(y_i|x_i) = \frac{\exp(x_i)}{\sum_{i=1}^{c} \exp(x_i)} \qquad \text{Equation 4.1}$$

where x_i is a vector of dimension c representing the un-normalized scores for the sample i.

The parameters of the network are determined through supervised training by minimizing the negative log likelihood over the training data. The loss function is defined as follows:

$$L(\mathrm{w}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} y_k^{(i)} \log(\hat{y}_k^{(i)}) \qquad \text{Equation 4.2}$$

where C is the length of vector of one-hot encoding of the semantic labels (i.e. number of classes). By comparing the true label vectors $y_k^{(i)}$ and the predicted label vectors $\hat{y}_k^{(i)}$ for n training samples, the loss function quantifies the misclassification error. The optimization problem is solved using stochastic gradient descent (SGD) (Bengio et al., 2015) with momentum α and learning rate ϵ from a small subset of the training data called a mini-batch. The decay learning rate ϵ_d and the batch size parameters are user-defined.

The learned parameters (weights and biases) are computed using backpropagation with gradient descent by calculating the derivative $\frac{\partial L}{\partial w_i}$ of the loss function L with respect to every parameter w_i. Adopting the notation in (Bergado et al., 2016), the weights in this work are updated by the equations considering the decay learning rate ϵ_d :

$$\Delta \mathrm{w}(\tau) = -\epsilon(\tau)\frac{\partial L(\tau)}{\partial \mathrm{w}(\tau)} + \alpha \Delta \mathrm{w}(\tau - 1) \qquad \text{Equation 4.3}$$

$$\epsilon(\tau) = \frac{\epsilon_0}{1 + \epsilon_d \tau} \qquad \text{Equation 4.4}$$

ϵ_0 is the initial learning rate, τ is the current epoch of the training phase. We mitigate overfitting through dropout, where a percentage d_r of the neurons and their connections is turned off. The parameter is set in the range of [0, 1]. We also use early stopping. This is whereby during training, the algorithm is run until the error on the validation set does not improve for a given number of epochs, e_n. Thirdly, L2 parameter norm penalty penalizes parameters deviating from zero. The resulting cost function after adding L2 regularization parameter is given in Equation 4.5 as:

$$J(\boldsymbol{w}) = L(\boldsymbol{w}) + \lambda \parallel \boldsymbol{w} \parallel^2 \qquad \text{Equation 4.5}$$

Where λ is the L2 regularization parameter and $\parallel w \parallel^2$ is the weight norm of the weight vector. Table 4.1 presents a description of CNN hyper-parameters.

Table 4.1: Definition of some hyper parameters adapted from (Bergado et al., 2016)

Hyper parameter	Description
m	Maximum span of the contextual neighbourhood to where the CNN is extracting the spatial contextual features
f	Size of the contextual patterns that can be learned by the CNN
h	The number of the hierarchical levels in the extraction of the spatial-contextual features
t	Complexity of the classification rule to map the spatial-contextual features to land-cover classes.

Some preliminary analysis was done on a small subset from Tile 2 measuring 501×501 pixels to determine the values of the learning and regularisation parameters to use. Accuracy assessment is carried out on the whole image tile. Figure 4.2 illustrates the subset and the corresponding reference data.

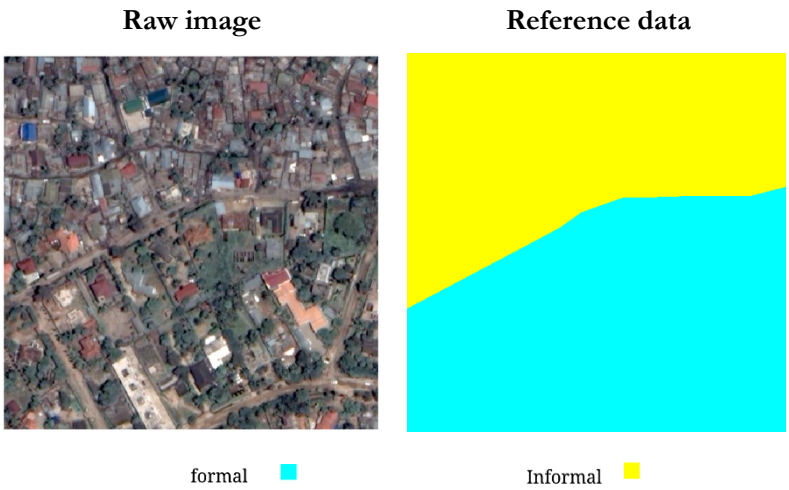


Figure 4.2. A subset used to derive the hyper-parameter values

A patch size of 65 and training set of 200 samples was used. The CNN had two convolutional layers and one fully connected layer. The network was trained using stochastic gradient descent over 200 samples whereas the learning and regularisation parameters were tuned over 200 held out validation samples. The overall accuracy of the network was evaluated over the whole image tile (251001 samples). The values of learning and regularisation parameters are shown in Table 4.2 while the CNN configuration used is presented in Table 4.3.

Table 4.2 Learning and regularisation parameters

Hyper-parameter	Values
Learning rate ϵ	(0.01,0.001)
Momentum α	0.9
Learning rate decay ϵ_d	(0.001,0.0001)
Early stopping patience e_n	(50)
Max number of epoch	1000
Weight decay λ	[("l2",0.001),("l2",0.0001)]
Dropout rate d_r (D1, D2)	(0.0, 0.5)

Table 4.3: CNN configuration

Hyper-parameter	Values
Layers[a]	I-C-A-P-D1-C-A-P-D1-F-D2-O
Nonlinearity used in A and F	$RELU$
Nonlinearity used in O	$softmax$
Width of F	128
Patch size m_s	65
Number of filters k	8
Kernel dimension f	7
Pooling size p	2

Key:

[a] Layer notation: I = input, C= Convolution, A = Activation, P = pooling, F=Fully Connected Layer, O = Output, D1 = dropout in the convolution stage, D2 = dropout in the fully connected layers. Weights are initialized using normalized initialization (Glorot & Bengio, 2010). The convolution stride is one while the pooling stride is two.

The combination of best parameters determined by hold-out cross-validation is used to train the CNN, followed by full image classification of the subset. The selected learning and regularisation parameters are presented in Table 4.4. These values are kept constant in all CNN experiments.

Table 4.4: List of final learning and regularisation parameters

Hyper-parameter	Values
Learning rate ϵ	0.001
Momentum α	0.9
Learning rate decay ϵ_d	0.001
Early stopping patience e_n	50
Max number of epoch	1000
Weight decay λ	0.0001
Dropout rate d_r (D1, D2)	(0.0, 0.5)

The parameters that are varied are patch size, number of kernels, dimension of kernels, number of convolutional layers and number of fully connected layers because they have an influence on the image feature learning.

Table 4.5: CNN configuration parameters and values used in all CNN experiments

Parameter	values
Layers[a]	I-(C-A-P-1)xC_n-(F-D2)xF_n-O
Nonlinearity used in A and F	$relu$
Nonlinearity used in O	$softmax$
Pooling size, p	2
Width of F	128

Table 4.6: A summary of the values used during CNN sensitivity analysis. The main diagonal indicates the values tried out. The columns represent the experiment carried out.

Parameters	Patch size m_s	Number of filters, k	Kernel Dimension, f	Convolutional layers, C_n	Fully connected, F_n
Patch size m_s	**(65,99,129,165)**	99	99	99	99
Number of filters k	8	**(8,16,32,64)**	8	8	8
Kernel dimension f	7	5	**(7,17,25)**	7	7
C_n	2	2	2	**(2,3,4)**	2
F_n	1	1	1	1	**(1,2,3)**

Key:

[a] Layer notation: I = input, C= Convolution, A = Activation, P = pooling, F=Fully Connected Layer, O = Output, D1 = dropout in the convolution stage, D2 = dropout in the fully connected layers. Weights are initialized using normalized initialization (Glorot & Bengio, 2010). The convolution stride is one, while the pooling stride is two. Border mode "same" is used for all the convolution layers, whereby the output feature map has the same size as the input just after the convolution layer.

We exclude $\left(\frac{m_s}{2}\right) - 1$ border pixels from all the sides of the tile when selecting samples during both training and testing, where m_s is the patch size. This because when samples are being picked from the tiles, these border pixels are padded with zeros and result in misclassification at inference time. This is done in all CNN experiments.

Patch size experiment

We start the CNN hyper-parameter optimization tests by evaluating the influence of varying the patch size on the classification results. Training of the network is carried by stochastic gradient descent over 2160 samples. The values of the learning and regularisation parameters are presented in Table 4.4 while the CNN configuration is described in Table 4.5. The values of m_s tried are 65, 99, 129 and 165. The CNN has two convolutional layers and one fully connected layer. The value of f used is five. A summary of the values used is presented in Table 4.6.

Dimension of kernel experiment.

We determine the influence of the dimension of the kernel (filter) on the overall accuracy of the network. We vary the values of f between 7, 17, and 25. The learning and regularisation parameters in Table 4.4 are used. The network is trained using stochastic gradient descent over the same sample set of 2160. We carry out a full image test over the three tiles to determine the overall accuracy of the network. We present the CNN configuration in Table 4.5. The value of m_s is set to 99. The rest of the parameter values are presented in Table 4.6.

Number of kernels experiment

We evaluate the effect of varying the number of kernels on the accuracy of the classification. In this experiment, we use the same training set of 2160 samples, drawn over three tiles to train the CNN using stochastic gradient descent. The value of m_s is fixed at 99. We vary the values of k between 8, 16, 32 and 64. The learning and regularisation parameters presented in Table 4.4 are used here. The same CNN, having two convolutional layers and one fully connected layer is used. Its description is presented in Table 4.5. We carry out a full image test over the three tiles to evaluate the overall accuracy of the network. A summary of all values used in the experiment are shown in Table 4.6.

Number of convolutional layers experiment

The effect of varying the number of convolution layers was also studied. Since the value of m_s was set as 99 and f set as seven, we evaluated up to four convolutional layers. The CNN carries out max-pooling with $p = 2$ and stride, $s = 2$. Therefore, the size of the feature map after the fourth CNN layer becomes less than the kernel dimension f. The value of C_n is varied between two, three, and four. The number of the fully-connected layers is maintained as one. The same training sample of 2160, drawn from the three tiles is used. The learning and regularisation parameters that are used are in Table 4.4. For the CNN configuration, an overview is presented Table 4.5. A summary of all values used in the experiment are shown in Table 4.6. A full image tests on the three image tiles is carried out to determine the overall accuracy.

Number of fully connected layers

Finally, we carry out an experiment to investigate the effect of varying the number of the fully connected layers between one, two and three. The value of m_s used is 99, and the value of f is seven. The number of convolutional layers is set to two. A training set of 2160 is used to train the CNN using stochastic gradient descent. The learning and regularisation parameters presented in Table 4.4 are used. A full image test is carried out on the three tiles. Details of the CNN configuration are presented in Table 4.5 and a summary of all values used in the experiment are shown in Table 4.6

4.1.2. GLCM window experiment

In this setup, samples are drawn from "formal" and "informal settlement" classes. We use Support Vector Machine classifier (SVM) with a radial basis function (RBF) kernel as the state of the art classifier. Classification using only the spectral bands as inputs is first done. Next spatial contextual features are extracted using Grey Level Co-occurrence Matrix (GLCM) and used in the classification (Haralick et al., 1973). GLCM variance is computed using Equation 4.6 as follows:

$$f = \sum_i \sum_j (i-\mu)^2 p(i,j) \qquad \text{Equation 4.6}$$

where $p(i,j)$ is the (i,j)th entry in a normalized gray-tone spatial dependence matrix, i and j are gray tones of neighbouring pixels.

Table 2.1 shows the parameters used when extracting the GLCM features. In GLCM-1, GLCM variance is extracted considering one direction according to (Kuffer, Pfeffer, Sliuzas, & Baud, 2016). In GLCM-4, the average of GLCM extracted over four directions is used. For the training of the SVM, we used hold out cross validation to determine the regularization parameter, *C* and the spread of the RBF kernel, *gamma*. We generate a logarithmically spaced vector of 25 elements for both parameters i.e., *C*= [1, 1000] and *gamma* = [0.0001, 1], resulting in 625 combinations. We carry out experiments to investigate the effect of varying window size of GLCM features on the classification result. Just as in the CNN experiments, the number of border pixels excluded all around the image tile is given by $\left(\frac{w_s}{2}\right) - 1$, where w_s is the window size in pixels. This is for the same reason that during GLCM extraction, the pixels at the borders are padded with zeros, likely resulting in misclassification.

Table 4.7: Description of parameters used to extract GLCM

GLCM Variance	**GLCM-1**	**GLCM-4**
Shift and lag	(1,1)	(0,1),(1,1),(1,0),(1,-1)
Window size, w_s	65, 99 , 129, 165	65, 99, 129, 165

4.1.3. Varying training sample size: SVM vs (SVM + GLCM)

We carried out an experiment to evaluate the effect of varying the size of the training samples on SVM, SVM+GLCM-1 and SVM+GLCM-4. Three different training sample sets from the three tiles. The size of the training sets was 1080, 2160 and 3060. As mentioned, the aim of this set of experiment was to evaluate the effect of varying the training set on these approaches. For both SVM+GLCM-1 and SVM+GLCM-4, a window size of 165 was used.

4.1.4. Varying training sample size: CNN

We carried out an experiment to evaluate the effect of varying the size of the training samples on CNN. The value of patch size, m_s was set as 165. The CNN architecture described in Table 4.5 is used, whereby two convolutional layers and one fully connected layer. However, the first convolutional layer has 32 kernels of dimension 25×25, whereas the second convolutional layer has 64 kernels of dimension 17×17. The learning and regularisation parameters that were used are the same as the ones shown in Table 4.4. A full image test over the three tiles is carried out to determine the overall accuracy of the classification.

4.2. Informal settlement vs other combined classes

As stated earlier, the original reference dataset comprises of four classes namely "Informal", "formal", "Vacant/Agriculture" and "other urban". In our first instance of experiments, we considered only "informal" and "formal" classes. However, we instead chose to merge the classes "formal", "Vacant/Agriculture" and "other urban" into one class called "other". This was done to support the aim of the experiments to distinguish "informal" from other unwanted classes such as other settlement types, vegetation and open spaces. Figure 4.3 shows the reference data for the three tiles-Tile 1, Tile 2 and Tile 3.

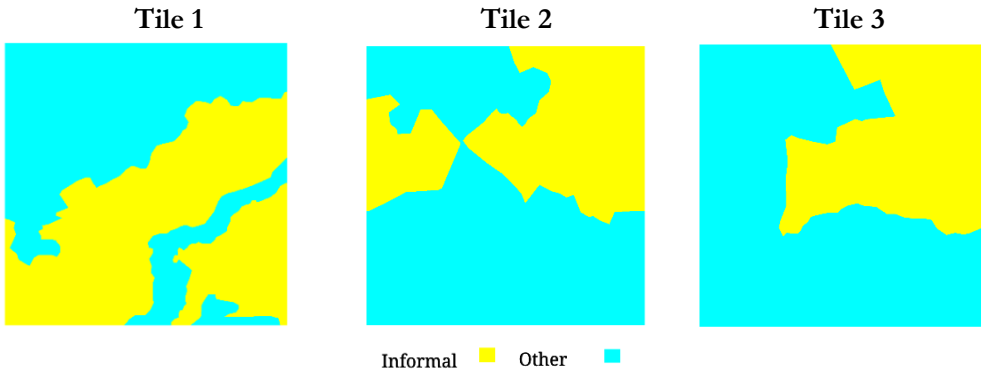


Figure 4.3: Reference data with two classes for three tiles-Tile 1, Tile 2 and Tile 3.

4.2.1. Varying convolutional layers vs varying the training sample size

In this experiment, we set out to determine whether a relationship exists between varying the training set and varying a CNN hyper-parameter simultaneously, and the effect on the result of the classification. To this aim, we vary the number of convolutional layers simultaneously with the training set size. The size of the training samples is varied from 1080, 2160, and 3060. These are drawn from the three tiles (i.e. Tile 1, Tile 2 and Tile 3) using stratified sampling and normalized in the range [0, 1]. For the CNN, training is conducted using stochastic gradient descent applying learning and regularisation parameters provided in Table 4.4. The CNN configuration that was used is the same as shown in Table 4.5. However, the difference is that a patch size of 165 is used. The CNN layers are varied from $C_n = 2, 3, 4, 5$ and 6. In each layer, there are eight kernels of dimension 7×7. For the fully connected layer, we use a value of $F_n = 1$.

4.2.2. Comparison of CNN vs SVM+GLCM

A comparison between the classification performance of the designed CNN and SVM+GLCM is conducted. From the experiments conducted in Section 4.2.1, we determine the training set size that provides the best CNN results (i.e. 3060). These classification results are compared to SVM+GLCM-1(i.e. for GLCM features extracted in one direction) and SVM+GLCM-4 (i.e. where GLCM features are extracted in four direction) whereby a window size of 165 was used during extraction. A full image test of the tiles is carried out to evaluate the accuracy of each method.

4.3. Exploration of the learned features vs extracted features

We set out to analyse and compare the extracted GLCM features and the learned CNN features from our experiments. The CNN is trained end-to-end and a softmax defined in Equation 4.1 is used to give the posterior class probabilities and the features are automatically learned. On the other hand, the GLCM features are extracted from the data. The classifier used, SVM with RBF kernel, is a state of the art machine learning classifier. We carried out a set of experiments to understand the utility of CNN learned features. We further exploited the possibility of combining the GLCM extracted features with the CNN learned features. We make use SVM with RBF kernel as the baseline classifier.

We make use of the CNN with five layers that was defined in Section 4.2, and trained over a sample set of 3060 with stochastic gradient descent and consider it as the first model. A second CNN is defined with the same configuration as the first CNN, then the weights learned from the first CNN are loaded onto the second CNN. The whole image tile is fed as an input to the second CNN. Feature maps are extracted after each dropout layer. Although dropout layer has been included, the actual dropout takes place during training and not during testing. The CNN pooling layers have a subsampling factor of s, where s is the stride of the pooling. In order to concatenate the feature maps and to extract training samples from the same locations, bilinear interpolation is carried out (Hariharan, Arbel, & Girshick, 2015). If n is the number of convolutional layer, then, the feature map needs to undergo a bilinear interpolation with a scale factor of s^n.

Figure 4.4 is a sketch that illustrates how CNN features are obtained and concatenated. We first use the features from each layer separately to carry out classification using SVM with RBF kernel. Next, we concatenate the first five feature maps and carry out classification. Finally, we concatenate the combined CNN features with the GLCM-1 features and GLCM-4 features respectively and perform a classification using SVM with RBF kernel. A full image test of the three tiles is used for accuracy assessment.

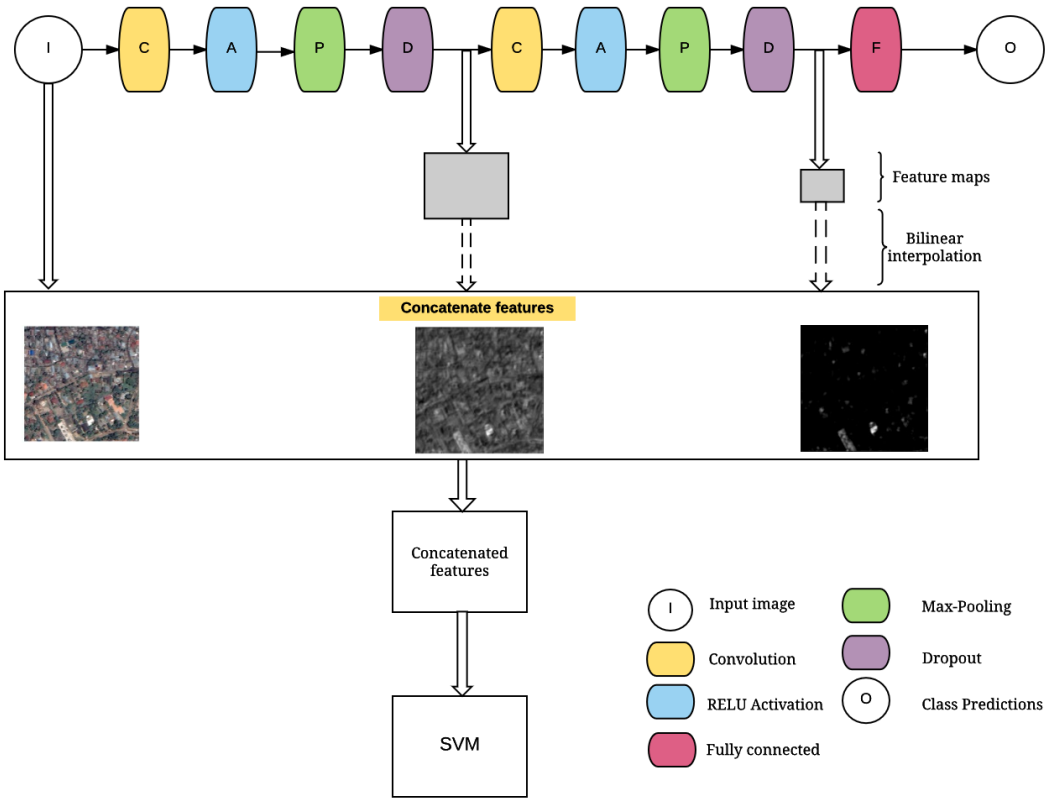


Figure 4.4: A schematic representation of the implementation of CNN+SVM

4.4. Accuracy assesment

Quantification of the accuracy of classification helps assign credibility to the classified map. We compute the global accuracy of the classification from the confusion matrix. The overall accuracy gives the rate of correctly classified pixels. It is derived from a confusion/error matrix created by comparing the classified pixel to the reference data. The producer and the user accuracies are calculated as well. The user's and the producer's accuracies were included to show the error contribution of each class. User's accuracy refers to the error of assigning a wrong label to a particular class. It is calculated by dividing the total number of correct pixels in a category by the total number of pixels classified into that class. On the contrary, producer's accuracy is the error of failing to assign a correct label to a particular class (Foody, 2002). Producer accuracy is the measure of the ability to classify a particular class while user accuracy is the measure of reliability of the classification (Congalton, 1991). We also carry out a visual quality assessment of the classified maps and compare the results among the methods used.

5. RESULTS AND ANALYSIS

In this chapter, results from the experiments conducted and their interpretation are presented. As described in Chapter 4, the preliminary experiments were to help in designing of the CNN. Experiments were conducted using the designed CNN and a comparison of the performance of CNN and SVM relying on GLCM carried out.

5.1. Preliminary experiments: informal settlements vs formal settlements

5.1.1. CNN hyper-parameter optimization

We start this Section by presenting the results of determining the learning and regularisation parameters. Table 5.1 shows the classification accuracy obtained by trying different combinations of learning rate, learning rate decay and weight decay. It is observed that the learning and regularisation parameters barely affect the overall accuracy of the network. For example the margin between the lowest and the highest classification accuracy is 0.63%. The learning and regularisation parameters in Set-up 6, were subsequently used for all CNN experiments. The value of ϵ, ϵ_d and λ were set as 0.001, 0.001 and ("l2", 0.0001) respectively. These are the values that affect the stochastic gradient descent algorithm.

Table 5.1: Overall accuracy from the learning and regularisation parameters experiments

Set-up	Learning rate, ϵ	Learning rate decay, ϵ_d	Weight decay, λ	Overall accuracy
1	0.01	0.001	("l2", 0.001)	94.49
2	0.01	0.001	("l2", 0.0001)	94.44
3	0.01	0.0001	("l2", 0.001)	94.43
4	0.01	0.0001	("l2", 0.0001)	94.47
5	0.001	0.001	("l2", 0.001)	94.85
6	0.001	0.001	("l2", 0.0001)	95.06
7	0.001	0.0001	("l2", 0.001)	94.85
8	0.001	0.0001	("l2", 0.0001)	94.94

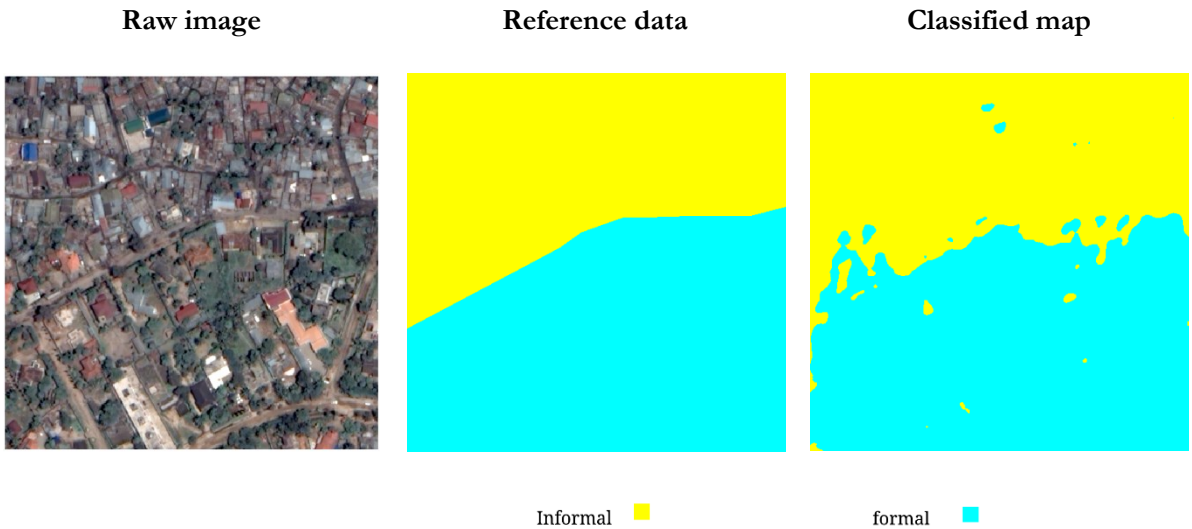


Figure 5.1: Classification result from the experiment for the determination of regularisation and learning parameters

Figure 5.1 presents the classified map from Set-up 6. The map is quite smooth, although there are some misclassifications. The confusion matrix presented in Table 5.2 shows that the user's accuracy of informal and formal settlement classes is high at 95.40% and 94.66% respectively. The promising results from using a subset encouraged the use of a bigger subset, and draw samples from different tiles.

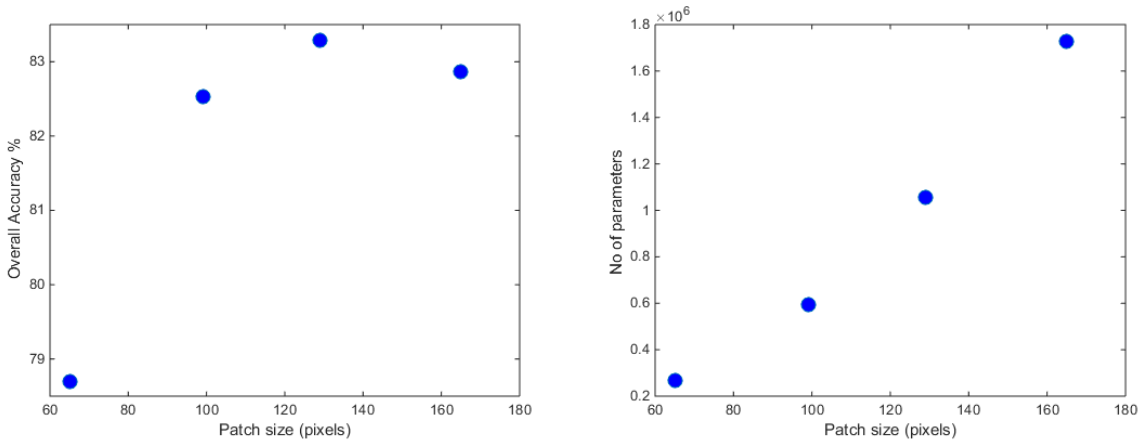
Table 5.2: Confusion matrix

	informal	**formal**	**User accuracy**
informal	129391	6225	**95.40**
formal	6163	109222	**94.66**
Producer accuracy	**95.45**	**94.60**	

Next, results of optimization of CNN hyper-parameters, which involved testing different values of the patch size, kernel dimension, number of kernels, number of convolutional layers and number of fully connected layers, are presented. In addition, the corresponding number of parameters in the CNN for each hyper-parameter value is presented to improve the understanding of the CNN hyper-parameter optimization.

Patch size

The patch size defines the size of the contextual window that is considered when assigning a label to the central pixel. The size of the contextual window around the pixel influences the label that is assigned (Farabet, Couprie, Najman, & Lecun, 2013). Figure 5.2 (a) illustrates the results of varying the patch size being fed to the CNN. Increasing the patch size increases the overall accuracy to a maximum of 83.29% at patch size of 129. Using a patch size of 165 causes the accuracy to fall slightly to 82.87%. This is despite that we would expect a large patch size to result in a better classification result because it provides more contextual information. The slight drop in accuracy can be attributed to the limiting factor of the fixed training sample size (2160) as the number of parameters grow as illustrated in Figure 5.2 (b).

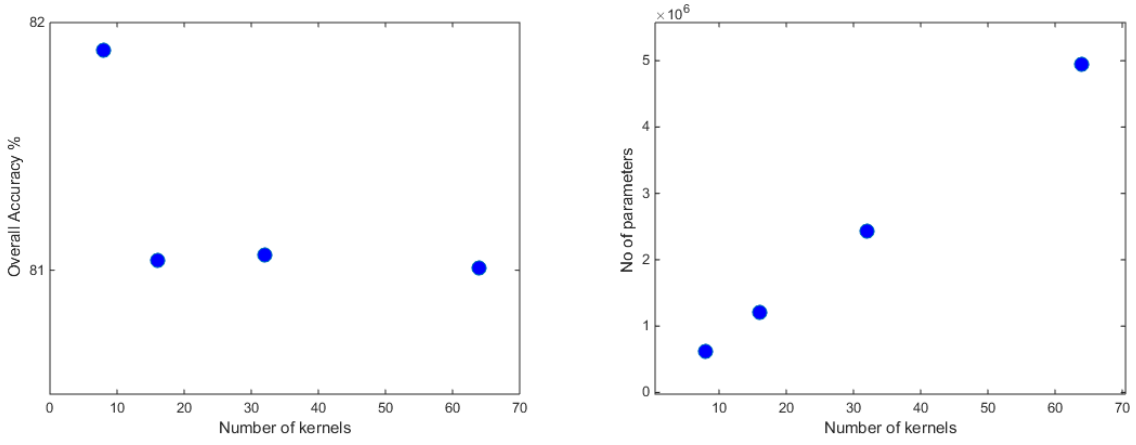


(a) (b)

Figure 5.2: Effect of varying the patch size on (a) the classification accuracy and (b) the number of CNN parameters

Number of Kernels

The number of kernels (filters) refers to the variety of spatial patterns that can be learned (Bergado et al., 2016) to distinguish land use classes. The results are presented in Figure 5.3. It can be seen in Figure 5.3 (a) that increasing the number of kernels results in a slight decrease in the overall accuracy from 81.89 % to 81.09%. Although the drop is less significant, it can be attributed to an increase in the number of parameters to be determined as the number of kernels are increased from 8 to 64 (Figure 5.3 (b)). Although we could expect a drastic drop in accuracy when the number of kernels increase, the slight drop can be attributed to the fact that a larger variety of informative features, which contribute to the better discrimination of the classes, are learnt. However, since the training set is fixed (2160), the CNN parameters are less optimally determined during training.

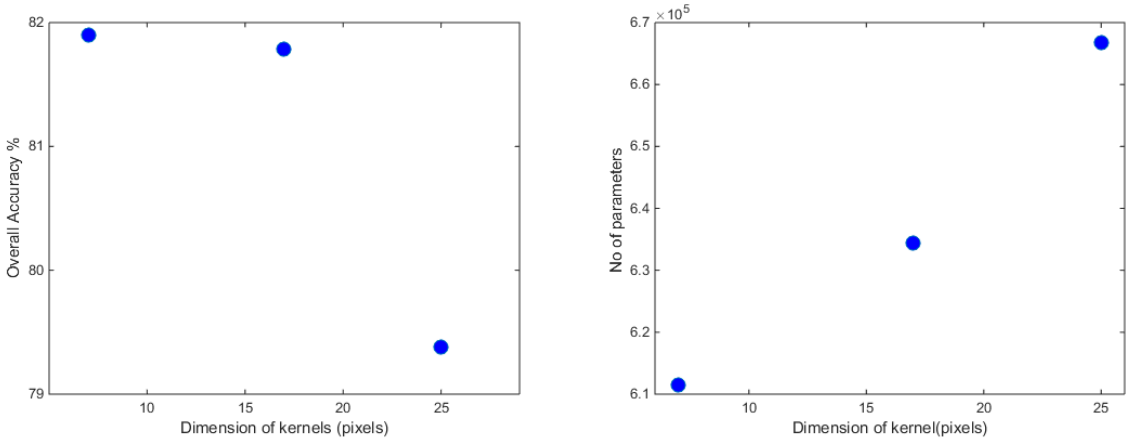


(a) (b)

Figure 5.3: Effect of varying the number of kernels on (a) the classification accuracy and (b) the number of CNN parameters.

Kernel dimension

We also present the results of varying the dimensions of the kernels. The size of the kernel defines the size of the patterns considered when distinguishing between classes of interest. It is also synonymous with the receptive field of a neuron. In other words, the size of the patterns in the image that will activate a particular neuron. In Figure 5.4 (a), we observe that increasing the dimension from 7 to 25 is accompanied by a fall in the overall accuracy by 2.52%. A large kernel size implies a larger receptive field. Consequently, the number of parameters to be optimized by the model during training increases as shown in Figure 5.4 (b). The CNN is less optimized because the training set sample was kept constant (2160).

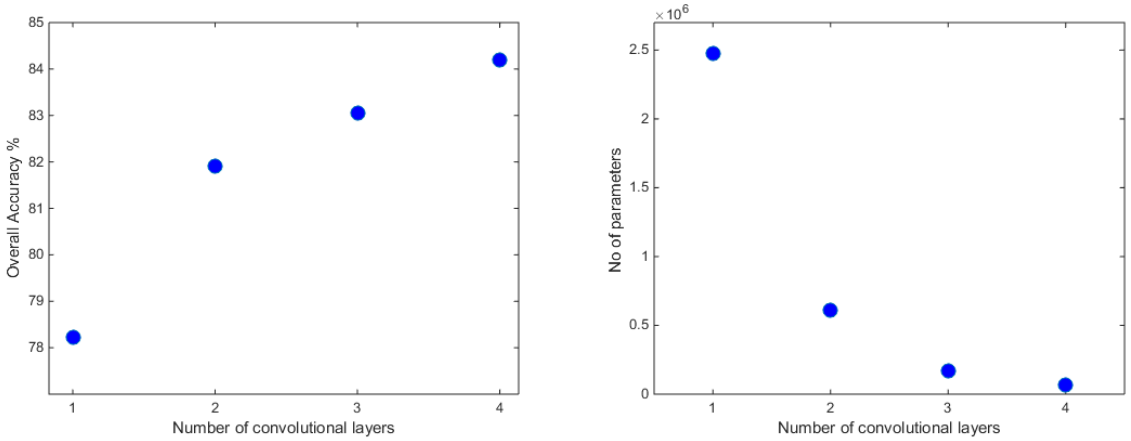


(a) (b)

Figure 5.4: Effect of varying the dimension of the kernel on (a) the classification accuracy and (b) the number of parameter

Number of convolutional layers

Increasing of the number of convolutional layers results in an increase in the overall accuracy of the classification as illustrated in Figure 5.5 (a). When two convolutional layers are used, the overall accuracy is 81.90% while four convolutional layers result in an accuracy of 84.19%. This translates to a significant increase of 2.29%.



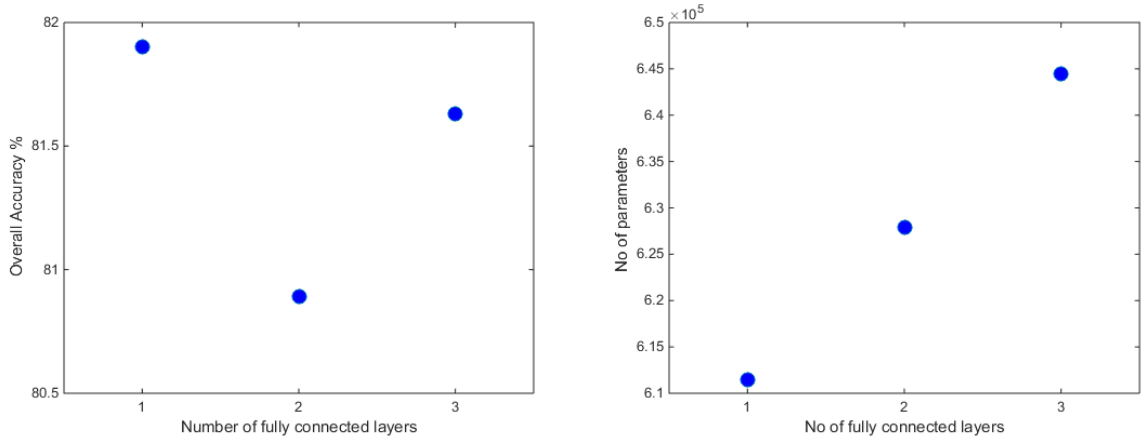
(a) (b)

Figure 5.5: Effect of varying the number of convolutional layers on (a) the accuracy of the classification and (b) the number of CNN parameters.

One of the reasons is that increasing the number of layers increases the level of abstraction of learned features. This means that the features enable better discrimination of the classes. Secondly, increasing the number of layers while keeping the number of filters constant is accompanied by a decrease in number of parameters. The number of parameters decreases from 2.5×10^6 to 64770 parameters for first and fourth convolutional layers respectively as shown in Figure 5.5 (b). The number of parameters decreases because the CNN has a pooling layer with a stride of two. This implies that the spatial resolution of the feature maps is reduced by a factor of two after each convolutional layer. This results in fewer parameters because the number of connections in the fully-connected layer is effectively reduced. This is an interesting observation because although the training sample size is fixed (2160), the accuracy still improves.

Number of fully connected layers

Lastly, we carried out an experiment to investigate the effect of varying the number of the fully connected layers. In Figure 5.6 (a), it is observed that increasing their number doesn't improve the classification. There is a drop of 1% when a second fully-connected layer is added and an increase of 0.74% when a third fully connected layer is added. Moreover, Figure 5.6 (b) illustrates that the corresponding increase in number of parameters is 33024. This could be one reason as to why increasing the number of fully connected layers has a slight effect on the classification performance of the network. Fully connected layers make use of all the features in the previous convolution layer to build features with stronger capabilities. In other words, fully connected layers develop the classification rule and their number determines the complexity of the classification rule (Bergado et al., 2016). We can postulate that in this scenario, the classification rule is not so complex to warrant many fully connected layers.



(a) (b)

Figure 5.6: Effect of varying the number of fully connected layers on the (a) accuracy of the classification and (b) number of CNN parameters.

5.1.2. GLCM window experiment

An experiment to investigate the effect of varying the window size of the extracted GLCM features on the classification result was conducted. As shown in Figure 5.7, increasing the window size results in a corresponding increase in the overall accuracy. The overall accuracy was computed over the three image tiles. A large window size implies that features extracted from a larger context (neighbourhood) are more descriptive than those extracted from a smaller context. This is sensible for detection of informal settlements because the class is more abstract and contains more than one land cover type. In addition, as can be seen from Figure 5.7, GLCM-4 performs slightly better than GLCM-1 for all the window sizes investigated except for 129. Whereas GLCM-1 is extracted at an angle of 45° (i.e. making it angular dependent), GLCM-4 is extracted by averaging textural features extracted over four directions namely: 0°, 45°, 90° and 135° (thus becoming rotationally invariant) (Haralick et al., 1973). While GLCM-1 is sensitive to patterns along one

direction (i.e. 45°or 225°), GLCM-4 is sensitive to patterns in any direction. This could explain why GLCM-4 features are better as compared to GLCM-1.

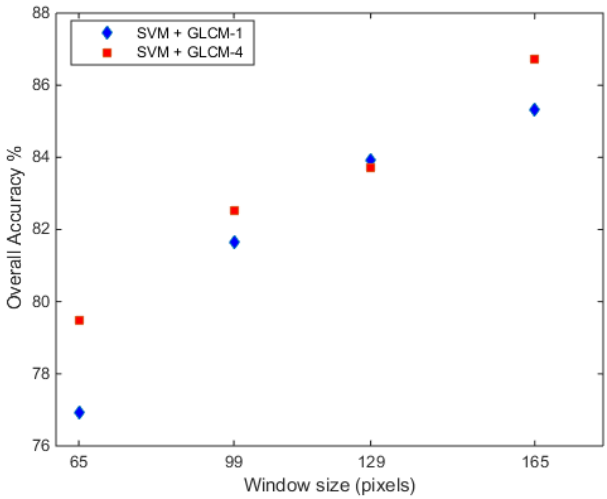


Figure 5.7: Effect of varying GLCM window size on the classification result, computed over three tiles.

5.1.3. Varying training sample size: SVM vs (SVM + GLCM)

We carried out an experiment to determine the effect of varying the size of the training sample on the classification accuracy of SVM, SVM+GLCM-1 and SVM+GLCM-4. The results are presented in Table 5.3.

Table 5.3: Classification accuracies for the SVM, SVM+GLCM-1 and SVM+GLCM-4 when varying training set.

	Patch 165	**SVM**	**SVM + GLCM-1**	**SVM + GLCM-4**
1080	Tile1	60.43	92.29	92.23
	Tile2	75.29	85.75	86.45
	Tile3	60.57	79.13	78.05
		65.43	**85.72**	**85.57**
2160	Tile1	59.67	92.16	92.83
	Tile2	76.99	86.98	87.99
	Tile3	61.97	76.89	79.34
		66.21	**85.34**	**86.72**
3060	Tile1	59.71	92.87	92.60
	Tile2	76.76	88.44	88.73
	Tile3	61.91	79.42	79.47
		66.13	**86.91**	**86.93**

Varying the size of the training sample barely influences the classification accuracy of SVM relying only on spectral bands. Similarly, it is observed that for both the SVM+GLCM-1 and SVM+GLCM-4, increasing the training sample size does not result in significant improvement of the overall accuracy.

Several insights can be drawn by comparing the performance of the three approaches. First, spectral information alone is insufficient to discriminate the two types of urban structures. This is shown by the

classification accuracy of SVM, which is at 65.43%, 66.21% and 66.13% for each of the training sets 1080, 2160 and 3060 respectively. However, adding spatial-contextual features improves the classification accuracy as can be seen from Table 5.3. For example, considering training set 1080, there is an increase of 20.29% when SVM+GLCM-1 is used over SVM. However, the difference on performance between GLCM-4 and GLCM-5 is not significant. We illustrate the results obtained from using a training sample of 3060 for SVM, SVM+GLCM-1 and SVM+GLCM-4 in Figure 5.8.

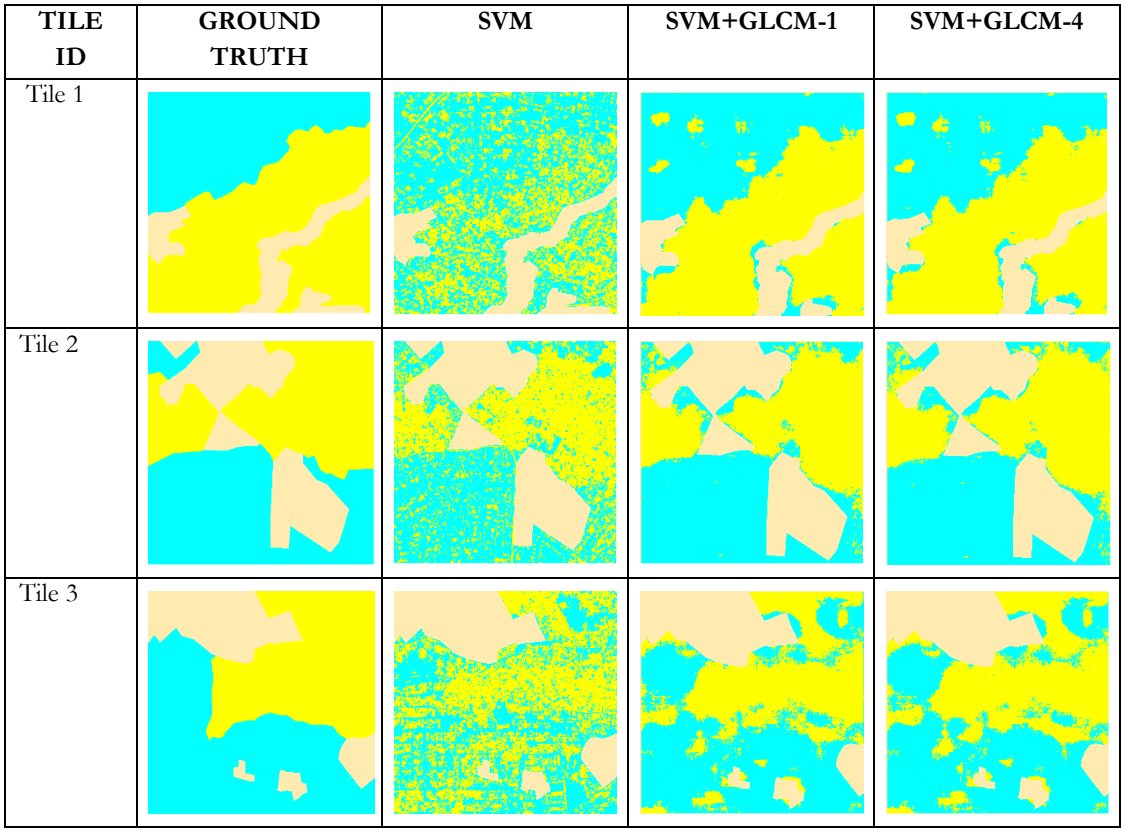


Figure 5.8: Classification results of SVM, SVM+GLCM-1 and SVM+GLCM-4.

The first column represents the ground reference data. The second columns presents the classification results of SVM. The third and fourth columns represent the results of SVM+GLCM-1 and SVM+GLCM-4 respectively. It is observed that classification of VHR relying on only spectral data produces noisy maps. This is evident across Tile 1, Tile 2 and Tile 3. The separation of the informal and formal settlement classes is not quite distinct. As expected, the classified images have a better visual quality when spatial-contextual features are used for classification. Considering SVM+GLCM-1, the classification of Tile 2 is quite good, with no noisy classification. However, some noise is present in Tile 1 and Tile 3. The classification maps from SVM+GLCM-1 and SVM+GLCM-4 are quite similar. The boundary of informal settlements is well captured when Tile 2 is considered for both SVM+GLCM-1 and SVM+GLCM-4. For Tile 3, the classified map is quite noisy, although location of the informal settlement is well captured. In all the approaches, classification results of Tile 3 are the lowest in contrast to the other two tiles. From visual interpretation of Tile 3, the informal and formal classes looked spectrally similar. Although the buildings in the formal settlement areas were spaced, there was limited presence of vegetation, which is a defining characteristic of an informal settlement. This ultimately posed a higher challenge for the discrimination of the informal and the formal settlement classes in this particular tile.

5.1.4. Varying training sample size: CNN

This set of experiment was done to see the effect of varying the size of the training set on CNN. Results of varying the training set are given in Table 5.4.

Table 5.4: Effect of varying training set for CNN

Sample size	Tile 1	Tile 2	Tile 3	Overall Accuracy
1080	65.68	79.39	57.46	**67.51**
2160	88.90	89.70	80.38	**86.32**
3060	90.00	88.84	83.51	**87.45**

The CNN performs rather dismally when a training sample set of size 1080 is used i.e. 67.51%. It is further observed that increasing the training set from 1080 to 3060 improves its accuracy of the classification with CNN from 67.51% to 87.45%. This is one evidence that CNN requires a large training set to optimally determine its parameters.

5.2. Informal settlements vs other combined classes

5.2.1. Varying convolutional layers vs varying training sample size

We also carried out an experiment to investigate the effect of varying the effect of varying the trainset against the number of convolution layers. This is significant because CNN performance varies significantly not only when training set size is varied but also when the hyper-parameter is varied. An illustration of the results in Figure 5.9 shows a trend whereby increasing the training sample size has a corresponding increase in performance of the CNN.

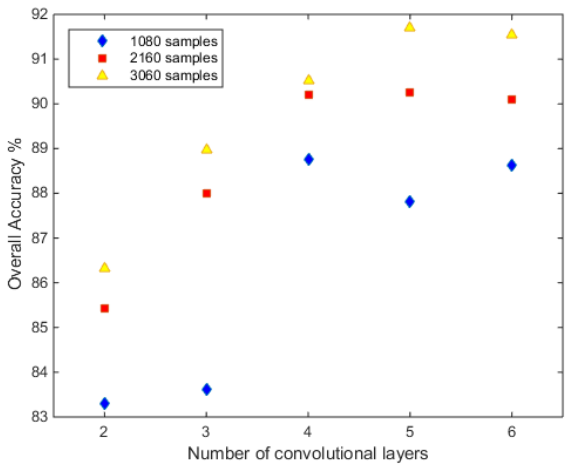


Figure 5.9: Effect of varying number of convolutional layers while varying the training sample size.

Looking at Figure 5.9, we see that the classification accuracy of the model when a training sample of 1080 and 2160 is used decreases beyond four convolutional layers. However, when a training sample size of 3060 is used, we see that the classification accuracy of the 5-layer CNN is quite high. It is also observed that for each training set size, there is a general trend whereby accuracy increases with each additional layer. However, going beyond four convolution layers does not significantly improve the overall accuracy of the classification. The insight derived from this experiment is that when determining the best CNN hyper-parameters, it is important to vary the training set size whilst varying the hyper-parameter values in order to strike a balance between the hyper-parameter value and the size of the training set.

5.2.2. Comparison of CNN vs (SVM+GLCM)

We carried out an experiment to investigate the detection of "informal" class against the "other" merged classes. One of the aim of this experiment was to demonstrate that CNN performs well when the training set is sufficient to allow for optimal determination of its parameters. As shown in Figure 5.10, SVM+GLCM-4 performs at par with SVM+GLCM-1 across the three tiles. In addition, both SVM+GLCM-1 and SVM+GLCM-4 have low classification accuracy in Tile 3. Performance for the CNN increases with each addition of a convolutional layer, with the highest classification accuracy being provided by CNN-5. We note that CNN-2 has lower classification accuracy than SVM +GLCM-4. However, when one layer is added, CNN-3 results in better classification accuracy for Tile 3 and Tile 4 only. For four, five and six convolutional layers, CNN outperforms SVM+GLCM-4 as seen in the classification accuracies of CNN-4, CNN-5 and CNN-6 respectively. In all the approaches, classification results of Tile 3 are the lowest in contrast to the other two tiles.

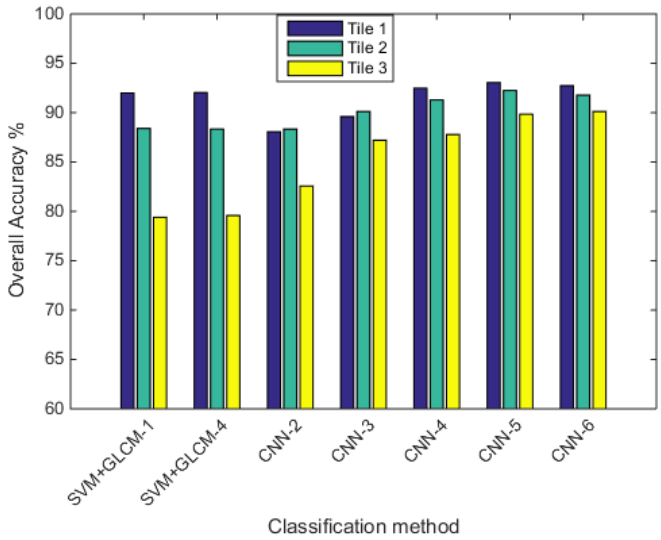


Figure 5.10: An illustration comparing the classification accuracy from CNN and SVM.

The classification accuracies for the respective tiles Tile1, Tile2 and Tile 3 are reported in Table 5.5. When six convolutional layers are used, there is a slight drop of overall accuracy in Tile 1 and Tile 2 of 0.32% and 0.47% respectively. However the classification accuracy of Tile 3 increases slightly by 0.26%. The classification accuracy of CNN is quite consistent across the three tiles as opposed to SVM relying on GLCM.

Table 5.5: Table presenting classification accuracies of investigated methods (SVM+GLCM and CNN)

TILE ID	SVM+GLCM-1	SVM+GLCM-4	CNN-2	CNN-3	CNN-4	CNN-5	CNN-6
Tile 1	91.99	92.03	88.06	89.60	92.48	93.05	92.73
Tile 2	88.40	88.34	88.33	90.11	91.28	92.24	91.77
Tile 3	79.40	79.58	82.57	87.20	87.78	89.85	90.11

The classified maps are presented in Figure 5.11. The columns represent the classified maps of Tile1, Tile2 and Tile 3 respectively. The first row indicates the ground reference maps while the subsequent two rows show the classified maps from using GLCM features. The fourth and fifth rows indicate results from using CNN with two (CNN-2) and five (CNN-5) convolutional layers respectively.

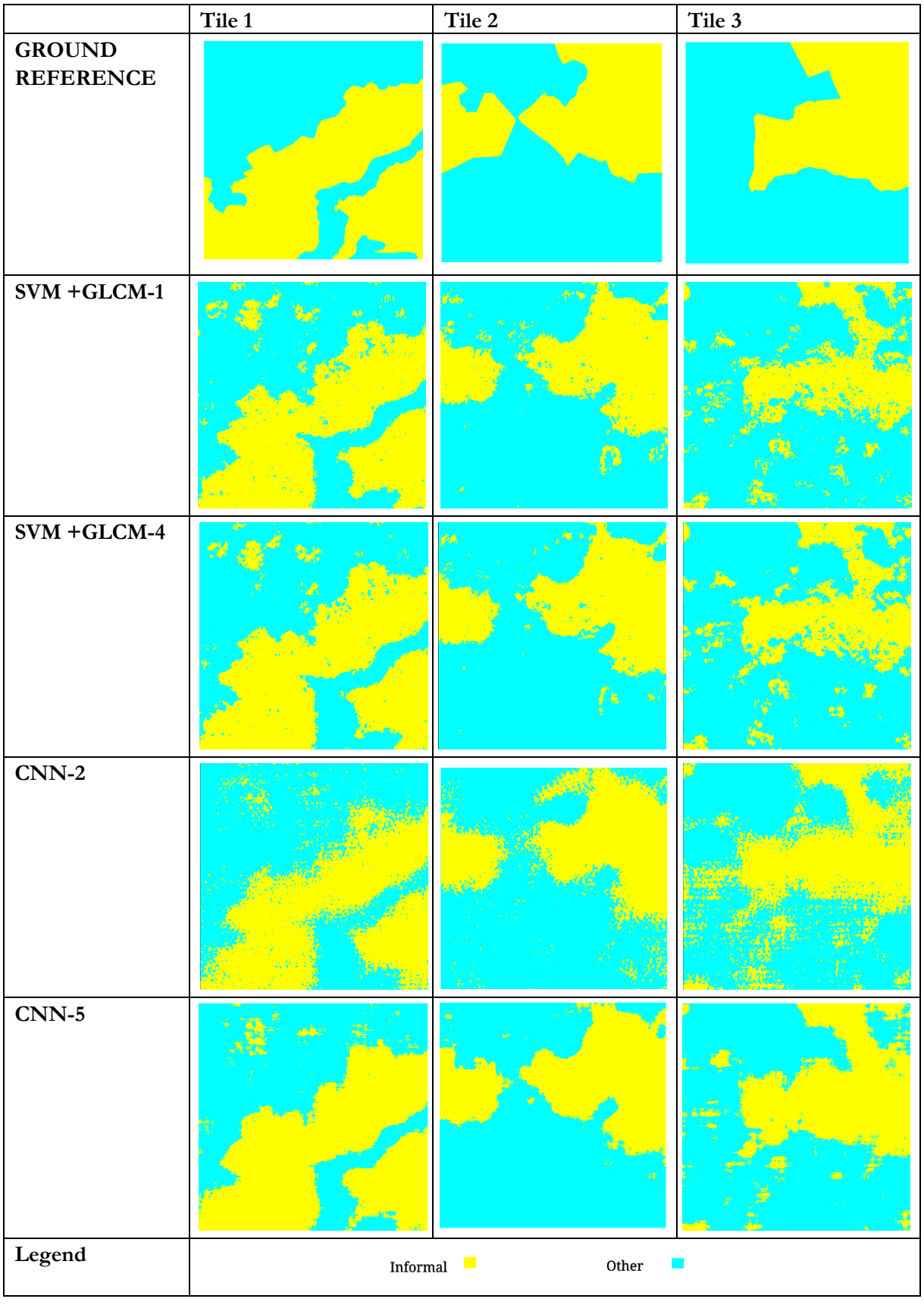


Figure 5.11: Classification maps from SVM relying on GLCM and CNN.

From Figure 5.11, we observe that boundary definition is much smoother in CNN-5 over CNN-2. Furthermore, it has less noisy classified maps than the latter. The difference between the classified maps by SVM+GLCM-1 and SVM-GLCM-4 is small. However, there are less noisy classifications in the latter than in the former. Considering CNN, using CNN-5 improves and reduces some of the misclassification especially in Tile 2. For both CNN and SVM+GLCM approaches, there is a clear misclassification in the north western corner of Tile 3, although it is much worse in the latter. Zooming in at the raw image, it is clear that it is an open field that falls within an informal settlement, see Figure 5.12 (c). This could be an example of existential uncertainty, whereby there is some doubt on the presence of the informal class in a given area. While still on Tile 3, it is evident that the extent of the informal settlement is better captured by the CNN approach as opposed to SVM +GLCM approach. Thus it can be inferred that spatial-contextual features that are learned by CNN are capable of discriminating the informal settlements from other urban settlement structures.

(a) Tile1 (b) Tile 2 (c) Tile 3

Figure 5.12: An illustration of regions in the raw image that are mostly misclassified. Shown in red boxes. The central area of Tile 1 has vegetation within an informal settlement. The north-western corners of Tile 2 and Tile 3 contain an open green field within an informal area.

5.3. Exploration of learned features vs extracted features

We present the results of using the feature maps from each convolutional layer to train a SVM with RBF kernel in Table 5.6 .In addition, the results of concatenating the feature maps derived from each of the convolution layers is presented. Finally, the results of concatenating the extracted GLCM features and the learned CNN features are presented in Table 5.7. The results show that the accuracy of the classification improves as features from higher layers are used. In addition, combining GLCM features and CNN feature maps results in a high classification accuracy.

Table 5.6: Use of CNN feature maps to train SVM

	Conv1	**Conv2**	**Conv3**	**Conv4**	**Conv5**
Area1	59.57	66.51	73.58	80.04	82.90
Area2	78.35	80.43	81.28	83.90	84.13
Area3	68.54	69.28	70.44	75.14	78.74

Table 5.7: Use of combined CNN feature maps and GLCM features to train SVM

	(Conv 1+…+Conv 5)	**Conv1+…+Conv5+GLCM-1**	**Conv1+….Conv5+GLCM-4**
Area4	84.41	89.70	90.28
Area 5	85.84	88.78	88.84
Area 6	79.49	81.86	82.65

Key:

1. (Conv 1+…+Conv 5) ➔ This is obtained by concatenating feature maps from the first five convolutional layers.
2. Conv1+…+Conv5+GLCM-1➔ This is obtained by concatenating features maps from first five convolutional layers and the GLCM-1 features.
3. Conv1+….Conv5+GLCM-4➔ This is obtained by concatenating feature maps from first five convolutional layers and the GLCM-4 features.

The CNN feature maps indicate regions and patterns of the input image that produce activation in the model. Feature maps in this work were generated by upsampling the actual feature maps, using bilinear resampling, to the original size of the input image to enhance the visualization process. An alternative method of visualizing the feature maps would have been to use a deconvnet (Zeiler & Fergus, 2014) where "switch variables" store the location of the dominant activations during pooling, and are used to reconstruct the feature maps at each convolution layer. A deconvnet approach would have provided a more accurate reconstruction of the feature maps. That notwithstanding, the feature maps obtained in this work sufficiently demonstrate the concept of hierarchy of features as described in CNN literature. Figure 5.13 is used to present the feature maps that were learnt from the CNN-5 when Tile 1 is used as an input.

The rows indicate the eight feature maps that are generated after each of the convolutional layers (represented by the columns). It can be clearly observed that low level features such as edges are more prominent in the 1st and 2nd layers (see Row 4, Col 1; Row 2, Col 1; Row 2, Col 2). However, there is an evolution of features from simple features to complex features when we consider layer four and five. See the following examples (Row 3, Col 4; Row 3, Col 5; Row 5, Col 5). It can be said that these feature maps represent regions instead of points, lines or edges. Giving it another perspective, we can say that the neurons in these layers (i.e. layer four and five) are activated by regions or abstract classes. They are able to respond to presence or absence of a particular class. Since we have two classes, namely "informal" and "other" settlements, the feature maps show activations for these two classes. We can basically distinguish the regions in the feature maps as belonging to either of these classes.

The lower layers of the CNN detect edges or local features whereas the higher layers detect more abstract representations. This owes to the fact that deeper layers in a convolution layer have a larger receptive field compared to the neurons/units in the lower layers (Bengio et al., 2015). They, therefore, can be said to effectively look into a larger area in the input image as compared to the lower layers. Visualizing a trained CNN model can give insight on how to design a CNN architecture (Zeiler & Fergus, 2014). It can help inform whether to use more CNN layers, or even the dimension of the kernels to use and influence the design of a CNN. The view that deep hierarchical features learn better discriminative features can be demonstrated by our experiments.

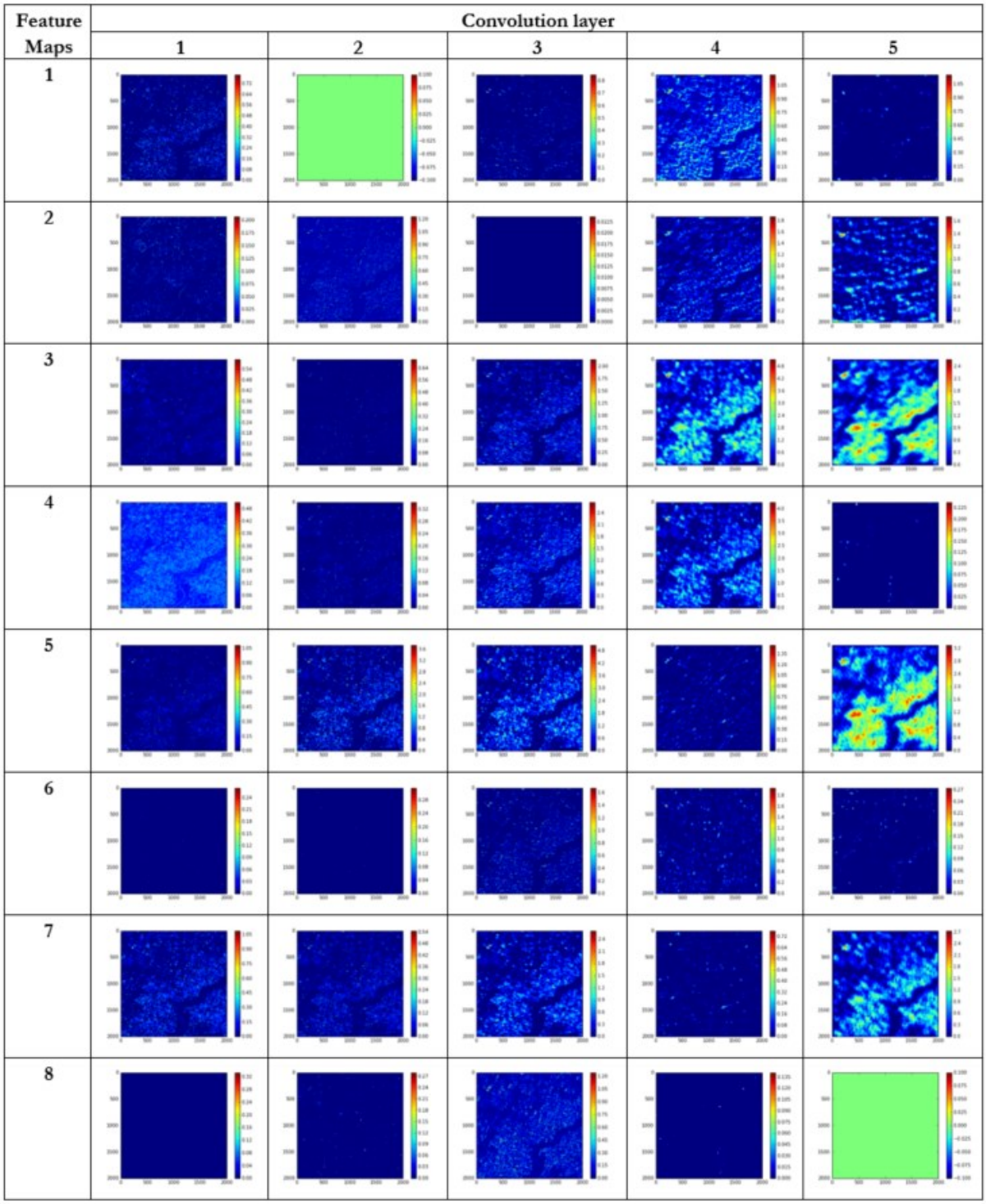


Figure 5.13: An illustration of 8 feature maps for Tile 1, derived from a CNN with 5 layers for each of the layers. The feature maps are upsampled through bilinear interpolation to attain a resolution of 2000×2000 pixels for visualization.

5.4. Accuracy assesment

In Table 5.8, the user and producer accuracy for the methods SVM+GLCM-1, SVM+GLCM-4 and CNN-2, CNN-3, CNN-4, CNN-5, and CNN-6 are presented. The accuracies are computed from the combined confusion matrices of Tile 1, Tile 2 and Tile 3. The results of this methods were discussed in Section 5.2.2. It is observed that CNN-5 has higher values for both "informal" and "other" classes. CNN-5 has a producer accuracy of 91.40 % and user accuracy of 88.22 %. This means that although the producer of the map can claim that 91.40 % of the time an area with informal settlements was classified as such, the user of the map will find that 88.22 % of the time the area they visit that the map says is informal will actually be informal. This is in contrast with SVM+GLCM-4 approach which has a producer accuracy of 94.39% and user accuracy of 75.63%. The error of commission is lower when CNN is used as opposed to SVM+GLCM. The lowest error of commission results from CNN-5 while the highest error of commission results from SVM+GLCM-1.

Table 5.8: Accuracy assessment for the methods SVM+GLCM-1, SVM+GLCM-4 and CNN, computed by combining the confusion matrix of Tile1, Tile 2 and Tile 3.

Approach	Overall Accuracy	Class	Accuracy		Error	
			User	Producer	Commission	Omission
SVM+GLCM-1	86.59	Informal	75.57	90.37	24.43	9.63
		Other	94.34	84.61	5.66	15.39
SVM+GLCM-4	86.65	Informal	75.63	90.44	24.37	9.56
		Other	94.39	84.65	5.61	15.35
CNN-2	86.32	Informal	84.71	84.71	15.29	15.29
		Other	87.38	87.37	12.62	12.63
CNN-3	88.97	Informal	81.56	91.38	18.44	8.62
		Other	89.66	87.58	10.34	12.42
CNN-4	90.51	Informal	85.29	91.14	14.71	8.86
		Other	94.17	90.11	5.83	9.89
CNN-5	91.71	Informal	88.22	91.40	11.78	8.6
		Other	94.17	91.92	5.83	8.08
CNN-6	91.53	Informal	87.70	91.43	12.3	8.57
		Other	94.22	91.60	5.78	8.4

Key:
The class "other" comprises of the combined classes "formal", "vacant/agriculture" and "other urban".

5.4.1. Errors due to uncertainty

Another important element of accuracy assessment is the existential and extensional uncertainty. Existential uncertainty relates to whether a phenomena is present in the location it is said to be. On the other hand, extensional uncertainty lies refers to lack of an exact definition of the boundary of a phenomena (Kohli, 2015). In the context of extraction of informal settlements from VHR, these two concepts were encountered. First, during creation of the reference data, we used visual interpretation to digitize the requisite classes. Some instances were encountered where there was a doubt as to whether an area was an informal settlement or not. A case in point is the presence of an open place within the informal settlement as illustrated in Figure 5.12 (c). Secondly, defining the exact delineation between an area that is informal and the area that is not was definitely a source of uncertainty (see Figure 5.14). Uncertainty in the location and boundary of a slum is likely to affect the accuracy of the reference data and the quality of the metrics calculated using it (Kohli, 2015).

(a) (b)

Figure 5.14: An illustration of extensional uncertainty. Although the classes have different morphological characteristics, a challenge lies in defining the exact extent of the classes when creating the reference data.

In this research, we were aware of these uncertainties and tried to mitigate them by using a reference dataset (Sliuzas, 2004; Sliuzas et al., 2016) as the first input to the visual interpretation process. In addition, morphological characteristics based on literature and described in were used to identify formal and informal areas. It would be desirable to quantify these uncertainties in future research.
In the next chapter, we discuss the results.

6. DISCUSSION

In this chapter, we present a discussion drawn from the analysis of our experimental results. A build up on some of the insights mentioned in the previous chapter is done. A comparison with existing research is done and in some cases, possible future research items are mentioned.

6.1. Utility of GLCM features

The results of using GLCM variance corresponds with the findings of Kuffer et al., (2016). GLCM variance is useful for extracting informal settlements from VHR because informal settlements exhibit lower values as opposed to formal settlements that have higher values (Kuffer, Sliuzas, Pfeffer, & Baud, 2015). The high variance values is because buildings in the formal areas are large, hence have higher contrast with surrounding as opposed to building structures in the informal areas that are small and hence portray lower contrast with the surroundings. This research experiments with both angular dependent GLCM (i.e. GLCM-1), and rotationally invariant GLCM (i.e. GLCM-4). The window size was the only parameter varied during GLCM extraction, and it has an effect on the contextual information that is derived. We note that there are other parameters that were kept constant during the extraction of GLCM features. These are the lag and shift which were both fixed as one, and the number of grey levels to use in the texture calculation set as eight. Determining the optimal values for these parameters could contribute to better performing GLCM features. The performance of GLCM-4 and GLCM-1 is almost similar and is quite competitive to CNN. Furthermore, the resulting classified maps are of good visual quality. This means that hand-crafted features can give competitive performance when carefully designed, although it is time consuming.

It is evident that the GLCM features have a fixed scale representation of the raw image. Specifically, there is only one level of representation of the extracted features. Extracting features at different window sizes and then concatenating them might provide more robust GLCM features. Since it has been shown in this work that GLCM variance features can generate competitive performance when carefully designed, it would be a possibility to explore ways of using information present in the GLCM features to improve the classification result. One of the ways is to use the extracted GLCM features with the spectral bands of an image and use them as input to a CNN as is the case in Sharma (2016). Exploration of other strategies of incorporating well designed hand-crafted features into the CNN classification pipeline need to be investigated.

6.2. Utility of CNN features

Conducting experiments with CNN features from the CNN with five layers provided several key insights, some of which are in tandem with literature in the computer vision domain. First, each of the convolution layer in the CNN-5 had eight kernels with a dimension of 7×7 pixels. It was evident that feature maps from higher layers (i.e. fourth and fifth layers) produced a high classification accuracy when used with SVM as opposed to using features from lower layers (i.e. first, second and third layers). However, the results of CNN+ SVM are lower than CNN-5. This might seem contrasting with the findings of (Athiwaratkun & Kang, 2015). However, on inspection of the model that they use, it is clear that their CNN layers had a higher number of kernels than that used in our experiment. For all their experiments, the lowest number of kernels used is 32 and the highest number is 64. We believe that using higher number of kernels would have resulted in similar results.

Another inference is that when carrying out end-to-end training using CNN, one of the main goals should be to ensure that the training sample is sufficient with respect to the number of the parameters in the model. That is why the CNN with eight kernels, as used in this work, would result in competitive results. On the other hand, increasing the number of kernels but maintaining the same size of the training set will

mostly reduce the classification accuracy of the CNN because of a resulting less optimally trained model. But looking on the flip side, the large variety of quality features would dramatically improve the classification of SVM based approach. This is because feature representation is significant in affecting the performance of SVM. Therefore we can postulate that it is possible to use features generated from a less optimized CNN model and use it to achieve competitive classification accuracy with a state-of-the art classifier such as SVM with RBF kernel.

In (Hariharan et al., 2015), the importance of understanding the utility of CNN learned features is espoused. Intermediate layers distinguish the local information well, whereas the higher layers disseminate the semantics better. The level of abstraction increases with increase in the number of layers of representation. CNN features are also used for classification with SVM and Random Forests (RF) in (Athiwaratkun & Kang, 2015). According to this paper, they find that SVM and RF trained with CNN features outperform the CNN and explain that it is because CNN uses the last layer (fully connected layer) to carry out its classification. They suggest that the fully connected layer is less optimal for training some classifiers. However, some work has also been carried out investigating the usefulness of the fully connected layer as a feature vector. For example, in the work by Razavian, Azizpour, Sullivan, & Carlsson (2014) experiments are conducted using a linear SVM and features from the first fully connected layer of "OverFeat" (Sermanet et al., 2013). They find that CNN learned features are significantly useful in visual recognition tasks. With adequate computation power and training data, a CNN can be optimized for a classification task for high performance. Further work needs to be done in investigating the utility of using fully connected layer for classifying satellite imagery using a state-of-the-art classifier.

6.3. Patch-based CNN

In our CNN experiments, patch-based (patch-wise) approach is used during the training and inference. This means that during training, location of pixel samples are identified. Then, patches around each of the pixels are extracted. While the locations of the pixels samples are randomly selected using stratified sampling, there is a high probability that several adjacent pixels are chosen. Consequently, some of the patches generated might be overlapping. This, in effect, creates redundant data and, in the long run, reduces the variety of the training samples that are used for training the model. One possible solution would be to set up a rule to ensure that non-overlapping patches are selected and used to train the CNN (although this would result in inadequate training samples if you consider extracting non-overlapping patches of 165×165 from a tile measuring 2000×2000).

Secondly, during inference, a patch around the pixel is considered before the label is assigned. Considering an image of 500×500 pixels, this means that this operation has to be repeated 250,000 times which demands more memory and processing time. This means that all the overlapping patches have to be sampled each time the central pixel is to be labelled. It would be possible to use patch-wise training but carry out image-based (image-wise) classification using techniques such as "shift-and-stitch" approach (Sermanet et al., 2013; Sherrah, 2016). We tried to implement the "shift-and-stitch" approach but because of time limitation, it will be done in later research works. An alternative approach would be to investigate fully convolutional neural networks (FCN) as introduced by (Long et al., 2015; Sherrah, 2016). The FCN is able to have an input that has arbitrary dimensions and generates an output of the same size and resolution. It carries out image-wise training and image-wise classification rather than patches. Image-wise implies that the loss function is minimised over whole image tile instead of a patch. Also, the deconvolution neural networks (DCN) would be another interesting CNN variant to explore (Noh et al., 2015; Volpi & Tuia, 2016; Zeiler et al., 2011). The DCN similarly allows for full image-wise training and testing.

The performance of the CNN is competitive in terms of overall accuracy and map quality in spite of the aforementioned limitations. It would be interesting to investigate strategies to improve the efficiency of the used CNN.

6.4. CNN hyper-parameter optimization

This research investigated the effect of varying the CNN hyper-parameters namely patch size, number of kernels, dimension of the kernels, the number of convolution layers, the number of fully connected layers. It emerged that a deeper CNN performed well as opposed to a shallow model. We demonstrated that one of the reasons for this was that the number of parameters decreased with each additional layer. This results might hold true because our CNN had a sub-sampling layer, with a factor of two. Without sub-sampling, the number of parameters would be large and it would be interesting to see whether the accuracy of the model improves with each additional layer. The performance of the CNN when varying the hyper-parameters was largely affected by the training sample size. We note that when determining the suitable values for CNN hyper-parameters for a particular task, it would be advisable to vary the training samples while varying the hyper-parameter values. This helps to strike a balance between the number of parameters in the CNN and its desired accuracy. Also, CNN with high number of convolutional layers had high classification accuracy because in a deep network, a hierarchy of simple to complex features is learnt, which improves discrimination of classes.

6.5. Training and Test sample size and quality

In this research, we used random stratified sampling to select the training samples. This implies that the number of samples from each class are chosen depending on the size of the class. Investigating the sampling strategies and their effect on the classification result. How to collect the samples from the image is important. Aside from the number of samples, the quality of the samples has a bearing on the classification result. It would be interesting to investigate a strategy whereby the locations from which samples are derived are defined in advance. This might result in some improvement in the robustness of the CNN. The size of the training sample influences whether the CNN will be optimally trained, consequently affecting the results of the classification results. A large training set ensures a better trained model, although takes longer time. During inference, all the pixels in the tiles are classified and evaluated. This eliminates bias from the computed accuracy measures.

6.6. Accuracy assesment using unsampled domain (Domain adaptation)

In this research, the classifiers were trained using samples drawn from all the available tiles. The model accuracy was evaluated by carrying out a full image test. This was one way to evaluate the performance of the classifiers by presenting them with unseen data. From the experimental results, it is observed that CNN outperforms SVM relying on GLCM. It would have been desirable to compare the performance of the approaches using tiles from which no training data were sampled. However, a better approach would be to evaluate a strategy of introducing training samples from the new tile (i.e. from which no samples were taken) so that a few but informative samples are used. One of such strategies is active learning presented in (Persello & Bruzzone, 2012). These experiments would help in evaluating the adaptability of CNN and SVM with GLCM. A successful strategy would make it possible to tune the CNN and use it for detection of any type of informal settlement types regardless of the geographical location. Although the urban settlement types have some common fundamental morphology characteristics, they are quite different depending on the geographical area (Kuffer et al., 2016). Thus, exploring the transferability of this approach to other environments would be a possible research step.

6.7. Final remarks

We have discussed the key insights that were evident in the course of this research. CNN is a very promising approach especially in the analysis of satellite imagery. Several limitations were identified which mainly were concerned with efficiency and possible solutions suggested. In addition to this, strategies to improve the accuracy of the classification and quality of the maps need to be explored.

7. CONCLUSION AND RECOMMENDATION

7.1. Reflection on the Objectives and the research questions

The main objective of this study was to investigate a deep feature learning approach based on convolutional neural networks (CNNs) for the detection of informal settlements. The available data was a Quickbird image taken over Dar es Salaam, Tanzania in 2007. We set out to develop a methodology based on CNN for detecting informal settlements from VHR satellite imagery. To this aim, we optimized the hyper-parameters of a CNN and trained the designed model in an end-to-end fashion. A detailed comparison of the performance of the CNN against state-of-the-art methods relying on handcrafted features (i.e. SVM+GLCM) was conducted. We also investigated the utility of CNN learned features, and the utility of combining them with GLCM features to train a SVM with RBF kernel.

Our experiments showed that SVM relying on GLCM features resulted in high classification accuracy. The quality of the maps was quite good, although there were noisy misclassifications. This showed that well designed hand-crafted features can exhibit competitive performance in a complex task involving classes with a higher level of semantic abstraction. Similarly, CNN had a high classification accuracy. However, it was evident that the CNN outperformed SVM+GLCM, especially when higher number of convolution layers, and a large training set was used. Using more convolutional layers enable the deep networks to learn a hierarchy of features, whereby the low layers learn basic features, while the higher layers learn more complex and abstract features. In other words, CNN learns spatial-contextual information features that help in discriminating complex classes. The result is improved classification accuracy and better quality of maps (i.e. considering location and extent of the classes). CNN requires large (adequate) training data for the optimal determination of the parameters of the network.

We also demonstrated that CNN feature maps have the utility of training a state-of-the-art classifier such as SVM, resulting in competitive classification accuracy. The learned CNN features can also be combined with well-designed hand-crafted features.

In summary, the methodology outlined in this work can be replicated to allow for extraction of informal settlements. CNNs trained in an end-to-end fashion can effectively learn complex, hierarchical and abstract features for land-use classification from VHR images.

The answers to the research questions raised at the onset of the study are thus provided:

1. How have the deep models been applied in the analysis of satellite imagery?

   In Section 2.2.2, it is described that Convolutional neural networks (CNNs) have recently gained prominence for land cover classification from high resolution (VHR) satellite and aerial imagery. Even new variants of CNN such as deconvolution neural networks (DCN), recurrent convolutional networks (RCN), and fully convolutional networks (FCN) have been mostly used for land cover classification tasks from VHR. However, there have been applications involving the use of CNN for land use classification. In this study, however, we used CNN for a land use classification task.

2. What are the building blocks of a CNN?

   In the CNN, the convolutional layers learn feature representations from data while the fully connected layers learn the classification rule from the learned features (Bergado et al., 2016). The theoretical background of CNN is discussed in Section 2.2.1. In addition, the mathematical formulations, a description of the network hyper-parameters, and the learning and regularisation parameters is presented in Section building blocks of the CNN are described in detail in Section 4.1.1.

3. How should the classes be defined?

   The main objective of this study is the detection of informal settlements from VHR satellite images using convolutional neural networks. A VHR satellite image of Dar es Salaam, Tanzania was used. As a precursor, the concept of informality was expounded in Section 2.1. This provided the theoretical foundation for using visual image interpretation, relying on morphological characteristics to create the reference dataset. Use of morphology characteristics is reliable because the urban settlement types are fairly distinct. We used an available land use reference as an input to the visual interpretation process. In the preliminary experiments, we distinguished between two classes namely "informal", and "formal". The class "others/vacant/agriculture" was not considered. However, in the final set of experiments, the class "informal" was distinguished against a merged class of "formal, other, vacant/agriculture".

4. What is the effect of varying the hyper-parameters on the classification results?

   We observed the effect of varying CNN hyper-parameters to the classification results through the CNN optimization experiments described in Section 4.1.1. The results presented in Section 5.1.1 showed that number of convolution layers significantly influenced the classification results. This confirmed the hypothesis that deeper network results in better discriminative features being learned by the CNN. The patch size and the kernel dimensions were also sensitive hyper-parameters. It was also observed that although varying the value of CNN hyper-parameters influences the classification accuracy, the size of the training sample had an effect. It emerged that varying the training set and the value of the hyper-parameters is a good strategy when optimizing the CNN.

5. What consideration should be made when designing a CNN?

   Basically, the appropriate learning and regularisation parameters need to be determined. The CNN hyper-parameters that affect the quality of the learned features should be determined. These hyper-parameters include patch size, number of convolution layer, number of fully connected layers, the number of kernels, and the dimension of the kernels. The problem at hand should guide the choice of set of values to use when varying during hyper-parameter optimization. In addition the spatial resolution of the image will determine the values of the optimal CNN hyper-parameters. Section 2.2.1 and Section 4.1.1 provide some of the considerations when designing a CNN.

6. How do the methods compare in terms of accuracy and on previously unseen data?

   In this study, the classification approaches based on CNN and on SVM+GLCM were evaluated by calculating the overall accuracy across the three image tiles. A comparison of the overall accuracies was done. Furthermore, a qualitative comparison of the classified maps was done visually. The results are discussed in Section 5.2. CNN with more convolutional layers and adequate training data has higher overall accuracy, user's accuracy and producers across the three image tiles, as opposed to SVM relying on GLCM.

To conclude, the contributions of this thesis are:

i) Demonstrating that deep learning algorithms, CNN in this case, are suitable for complex land use classification tasks such as detection of informal settlements from VHR satellite imagery.

ii) Development of a methodology based on CNN to detect and map informal settlements from VHR satellite imagery.
iii) Demonstrating that CNN learned features alone, and when combined with GLCM features, can be used to train SVM with RBF kernel with competitive classification results.

7.2. Recommendations and future works

For future works, we recommend the following:

- Investigate full image training and testing for land use classification which is different from the patch-based approach used in this study using fully convolutional networks (FCN) or deconvolutional neural networks (DCN).
- Explore techniques of improving accuracy of the CNN by addressing the noisy misclassifications in the classified maps.
- Investigate the effect of data augmentation, and other techniques of expanding the training set size for classification of VHR satellite imagery using CNN.
- Investigate domain adaptation using CNN in a land cover/land use classification problem using remote sensing data and further, look at strategies for fine-tuning a CNN for landcover/landuse classification task.
- Lastly, classification results from GLCM were competitive to the CNN. To investigate how GLCM features can be used to accelerate/influence CNN training will be possible research direction.

LIST OF REFERENCES

Athiwaratkun, B., & Kang, K. (2015). Feature Representation in Convolutional Neural Networks. *arXiv:1507.02313 [Cs]*, 6–11. Retrieved from http://www.arxiv.org/pdf/1507.02313.pdf

Bengio, Y., Goodfellow, I., & Courville, A. (2015). Deep learning. *An MIT Press Book in Preparation*. http://doi.org/10.1016/S0022-3913(12)00047-9

Bergado, J. R. A., Persello, C., & Gevaert, C. (2016). A deep learning approach to the classification of sub-decimeter resolution aerial images. In *IEEE International Geoscience and Remote Sensing Symposium* (pp. 1516–1519). Beijing.

Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). Land Use Classification in Remote Sensing Images by Convolutional Neural Networks. *arXiv Preprint arXiv:1508.00092*, 1–11. Retrieved from http://arxiv.org/abs/1508.00092

Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, *37*(1), 35–46. http://doi.org/10.1016/0034-4257(91)90048-B

CS213n. (2016). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved January 12, 2017, from http://cs231n.github.io/convolutional-networks/

Deng, L. (2014). Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, *7*(3–4), 197–387. http://doi.org/10.1561/2000000039

Farabet, C., Couprie, C., Najman, L., & Lecun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1915–1929. http://doi.org/10.1109/TPAMI.2012.231

Foody, G. M. (2002). Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, *80*(1), 185–201. http://doi.org/10.1016/S0034-4257(01)00295-4

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, *9*, 249–256. http://doi.org/10.1.1.207.2059

Gueguen, L. (2015). Classifying compound structures in satellite images: A compressed representation for fast queries. *IEEE Transactions on Geoscience and Remote Sensing*, *53*(4), 1803–1818. http://doi.org/10.1109/TGRS.2014.2348864

Haralick, R., Shanmugan, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*. http://doi.org/10.1109/TSMC.1973.4309314

Hariharan, B., Arbel, P., & Girshick, R. (2015). Hypercolumns for Object Segmentation and Fine-grained Localization, 447–456. http://doi.org/10.1109/CVPR.2015.7298642

Hill, A., & Lindner, C. (2010). Modelling informal urban growth under rapid urbanisation. Retrieved from https://eldorado.tu-dortmund.de/bitstream/2003/27283/1/Dissertationsschrift_Hill_Lindner_Juni_2010.pdf

Hofmann, P. (2014). Defining Robustness Measures for OBIA Framework: A Case Study for Detecting Informal Settlements. In *Global Urban Monitoring and Assessment through Earth Observation* (pp. 303–324). CRC Press. http://doi.org/doi:10.1201/b17012-21

Hofmann, P., Strobl, J., Blaschke, T., & Kux, H. (2008). Detecting informal settlements from Quickbird data in Rio de Janeiro using an object based approach. *Object-Based Image Analysis*, (2001), 531–553. http://doi.org/10.1007/978-3-540-77058-9_29

Hu, F., Xia, G.-S., Hu, J., & Zhang, L. (2015). Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sensing*, *7*(11), 14680–14707. http://doi.org/10.3390/rs71114680

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for automatic human action recognition. Google Patents. Retrieved from https://www.google.com/patents/US8345984

Kironde, J. M. L. (2006). The regulatory framework , unplanned development and urban poverty : Findings from Dar es Salaam , Tanzania. *Land Use Policy*, *23*(4), 460–472. http://doi.org/10.1016/j.landusepol.2005.07.004

Kivinen, J. J., & Williams, C. K. I. (2011). Transformation equivariant Boltzmann machines. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6791 LNCS*(PART 1), 1–9. http://doi.org/10.1007/978-3-642-21735-7_1

Kohli, D. (2015). *Identifying and classifying slum areas using remote sensing*. University of Twente Faculty of Geo-

Information and Earth Observation (ITC). Retrieved from http://purl.org/utwente/doi/10.3990/1.9789036540087%5Cnhttps://www.itc.nl/library/papers_2015/phd/kohli.pdf

Kohli, D., Sliuzas, R., Kerle, N., & Stein, A. (2012). An ontology of slums for image-based classification. *Computers, Environment and Urban Systems*, *36*(2), 154–163. http://doi.org/10.1016/j.compenvurbsys.2011.11.001

Kombe, W. J., & Kreibich, V. (2001). Informal land management in Tanzania and the misconception about its illegality. In *ESF/N-Aerus Annual Workshop."Coping with Informality and Illegality in Human Settlements in Developing Countries", in Leuven and Brussels*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances In Neural Information Processing Systems* (pp. 1097–1105). Curran Associates, Inc. http://doi.org/http://dx.doi.org/10.1016/j.protcy.2014.09.007

Kuffer, M., & Barros, J. (2011). Procedia Environmental Sciences Urban Morphology of Unplanned Settlements : The Use of Spatial Metrics in VHR Remotely Sensed Images, *0*. http://doi.org/10.1016/j.proenv.2011.07.027

Kuffer, M., Barros, J., & Sliuzas, R. V. (2014). Computers , Environment and Urban Systems The development of a morphological unplanned settlement index using very-high-resolution (VHR) imagery. *Computers, Environment and Urban Systems*, *48*, 138–152. http://doi.org/10.1016/j.compenvurbsys.2014.07.012

Kuffer, M., Pfeffer, K., Sliuzas, R., & Baud, I. (2016). Extraction of Slum Areas From VHR Imagery Using GLCM Variance. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(5), 1830–1840. http://doi.org/10.1109/JSTARS.2016.2538563

Kuffer, M., Sliuzas, R., Pfeffer, K., & Baud, I. (2015). The utility of the co-occurrence matrix to extract slum areas from VHR imagery. The case of Mumbai, India. *2015 Joint Urban Remote Sensing Event (JURSE)*, 3–6. http://doi.org/10.1109/JURSE.2015.7120514

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://doi.org/10.1038/nature14539

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation ppt. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440. http://doi.org/10.1109/CVPR.2015.7298965

Lu, D., Hetrick, S., & Moran, E. (2010). Land Cover Classification in a Complex Urban-Rural Landscape with QuickBird Imagery Land Cover Classification in a Complex Urban-Rural Landscape with QuickBird Imagery. *International Photogrammetric Engineering & Remote Sensing*, *76*(10), 1159–1168. http://doi.org/10.1016/j.biotechadv.2011.08.021.Secreted

Luus, F. P. S., Salmon, B. P., Van Den Bergh, F., & Maharaj, B. T. J. (2015). Multiview Deep Learning for Land-Use Classification. *IEEE Geoscience and Remote Sensing Letters*, *12*(12), 2448–2452. http://doi.org/10.1109/LGRS.2015.2483680

Mehrotra, K., Mohan, C. K., & Ranka, S. (1997). *Elements of artificial neural networks*. MIT press. Retrieved from https://books.google.nl/books?id=6d68Y4Wq_R4C&printsec=frontcover

Noh, H., Hong, S., & Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. *Iccv*, *1*, 1520–1528. http://doi.org/10.1109/ICCV.2015.178

Paisitkriangkrai, S., Sherrah, J., Janney, P., & Hengel, A. Van Den. (2016). Semantic Labeling of Aerial and Satellite Imagery, *9*(7), 1–14. http://doi.org/10.1109/JSTARS.2016.2582921

Persello, C., & Bruzzone, L. (2012). Active Learning for Domain Adaptation in the Supervised Classification of Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*, *50*(11), 4468–4483. http://doi.org/10.1109/TGRS.2012.2192740

Pinheiro, P., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. *Proceedings of The 31st International Conference …*, *32*(June), 82–90. Retrieved from http://infoscience.epfl.ch/record/192577/files/Pinheiro_Idiap-RR-41-2013.pdf%5Cnhttp://jmlr.org/proceedings/papers/v32/pinheiro14.html

Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: An astounding baseline for recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 512–519. http://doi.org/10.1109/CVPRW.2014.131

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. http://doi.org/10.1016/j.neunet.2014.09.003

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). OverFeat: Integrated

Recognition, Localization and Detection using Convolutional Networks. *arXiv Preprint arXiv*, 1312.6229. Retrieved from http://arxiv.org/abs/1312.6229

Sharma, K. (2016). Classification of Mammogram Images by using CNN Classifier, 2743–2749.

Shekhar, S. (2012). Detecting slums from Quick Bird data in Pune using an object oriented approach. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXIX-B8*(September), 519–524. http://doi.org/10.5194/isprsarchives-XXXIX-B8-519-2012

Sherrah, J. (2016). Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery, 1–22. Retrieved from http://arxiv.org/abs/1606.02585

Sliuzas, R. V. (2004). Managing Informal Settlements: A Study Using Geo-Information in Dar es Salaam, Tanzania. *ITC Publication Series*, *112*. Retrieved from http://www.itc.nl/library/Papers_2004/phd/sliuzas.pdf

Sliuzas, R. V, Hill, A., Lindner, C., & Greiving, S. (2016). Dar es Salaam Land Use and Informal Settlement Data Set. Palisades, NY: NASA Socioeconomic Data and Applications Center (SEDAC). Retrieved from http://dx.doi.org/10.7927/H43T9F56

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. http://doi.org/10.1214/12-AOS1000

Tokarczyk, P., Wegner, J., Walk, S., & Schindler, K. (2013). Beyond hand-crafted features in remote sensing. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II-3/W1*(May), 35–40. http://doi.org/10.5194/isprsannals-II-3-W1-35-2013

UN-Habitat. (2015). Habitat III Issue Papers 22 – Informal Settlements. *United Nations Conference on Housing and Sustainable Urban Development*, *2015*(May), 0–8. http://doi.org/http://dx.doi.org/10.3402/gha.v5i0.19065

UN-HABITAT. (2003). Global urban observatory. Guide to monitoring target 11: Improving the lives of 100 million slum dwellers. Retrieved from http://mirror.unhabitat.org/pmss/listItemDetails.aspx?publicationID=1157

UN-HABITAT. (2012). Housing & slum upgrading. Retrieved July 30, 2016, from http://unhabitat.org/urban-themes/housing-slum-upgrading/

Vatsavai, R. R., Bhaduri, B., & Graesser, J. (2013). Complex settlement pattern extraction with multi-instance learning. In *Joint Urban Remote Sensing Event 2013, JURSE 2013* (Vol. 856, pp. 246–249). Sao Paulo: IEEE. http://doi.org/10.1109/JURSE.2013.6550711

Vatsavai, R. R., Bright, E., Varun, C., Budhendra, B., Cheriyadat, A., & Grasser, J. (2011). Machine learning approaches for high-resolution urban land cover classification. *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications - COM.Geo '11*, 1–10. http://doi.org/10.1145/1999320.1999331

Volpi, M., & Tuia, D. (2016). Dense semantic labeling of sub-decimeter resolution images with convolutional neural networks, 1–13. Retrieved from http://arxiv.org/abs/1608.00775

Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks arXiv:1311.2901v3 [cs.CV] 28 Nov 2013. *Computer Vision–ECCV 2014*, *8689*, 818–833. http://doi.org/10.1007/978-3-319-10590-1_53

Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. *Proceedings of the IEEE International Conference on Computer Vision*, 2018–2025. http://doi.org/10.1109/ICCV.2011.6126474

APPENDIX

Appendix A: CNN hyper-parameter optimization classification results

Table A- 1: Effect of varying the patch-size on the classification accuracy of the 3 tiles

Patch size	Tile 1	Tile 2	Tile 3
65	81.27	84.79	70.05
99	85.62	86.51	75.48
129	87.38	86.83	75.68
165	84.68	88.11	75.83

Table A- 2: Effect of varying the number of kernels on the classification accuracy of the three tiles

Number of filters	Tile 1	Tile 2	Tile 3
8	84.52	87.61	73.56
16	84.87	86.80	71.46
32	84.72	86.62	71.86
64	83.12	86.42	73.50

Table A- 3: Effect of increasing the kernel dimension on the classification accuracy of the three tiles

Kernel dimension	Tile 1	Tile 2	Tile 3
7	84.52	87.61	73.56
17	83	87.35	75.03
25	74.97	86.67	76.49

Table A- 4: Effect of increasing the number of convolutional layers to the classification accuracy of the three tiles

Number of Layers	Tile 1	Tile 2	Tile 3
1	85.28	84.88	64.5
2	84.52	87.61	73.56
3	85.67	87.80	75.72
4	85.90	87.89	78.69

Table A- 5: Effect of varying the number of fully connected layers on the classification accuracy of the three tiles.

Number of layers	Tile 1	Tile 2	Tile 3
1	84.52	87.61	73.56
2	80.21	85.78	76.7
3	85.14	86.76	73.00

Computed number of parameters in the network using the model. Summary () function in Keras

Table A- 6: Effect of varying the patch size on the number of CNN parameters

Patch size	Number of parameters
33	70642
65	267250
99	594930
129	1053682
165	1726450

Table A- 7: Effect of varying the kernel dimension on the number of CNN parameters

Dimension of kernel	parameters
7	611442
17	634482
25	666738

Table A- 8: Effect of varying the number of kernels on the number of CNN parameters

Number of filters	parameters
8	611442
16	1212258
32	2432706
64	4948866

Table A- 9: Effect of varying the number of convolutional layers on the number of CNN parameters

Convolution layers	parameters
1	2477098
2	611442
3	172218
4	64770

Table A- 10: Effect of the number of fully connected layers on the number of CNN parameters

Fully connected layers	parameters
1	611442
2	627954
3	644466

Appendix B: GLCM window experiments

Table B- 1: Effect of varying the GLCM window size on the classification accuracy

Patch-size	Tile id	SVM + GLCM1	SVM+GLCM2
65	Area1	72.44	78.99
	Area2	85.22	86.40
	Area3	66.96	72.67
99	Area1	76.49	86.55
	Area2	85.23	85.14
	Area3	72.14	72.95
129	Area1	77.08	86.45
	Area2	86.83	90.32
	Area3	71.87	80.58
165	Area1	80.85	89.95
	Area2	89.10	92.45
	Area3	72.31	83.87

Appendix C: Varying size of training set vs varying the number of convolution layers

Table C- 1: Effect of varying size of training set vs varying the number of convolutional layers

Training set	Tile ID	Number of convolution layers				
		2	**3**	**4**	**5**	**6**
1080	Area1	83.42	78.67	90.24	91.48	91.88
	Area2	87.77	89.51	89.36	88.87	89.39
	Area3	78.73	82.70	86.68	83.11	84.59
		83.31	**83.62**	**88.76**	**87.82**	**88.62**
2160	Area1	85.13	89.63	91.47	91.02	92.37
	Area2	88.45	88.59	90.26	90.79	91.46
	Area3	82.72	85.81	88.83	88.97	89.49
		85.43	**88.01**	**90.19**	**90.26**	**90.10**
3060	Area1	88.06	89.60	92.48	93.05	92.73
	Area2	88.33	90.11	91.28	92.24	91.77
	Area3	82.57	87.20	87.78	89.85	90.11
		86.32	**88.97**	**90.51**	**91.71**	**91.53**

Appendix D: Classification maps and feature maps

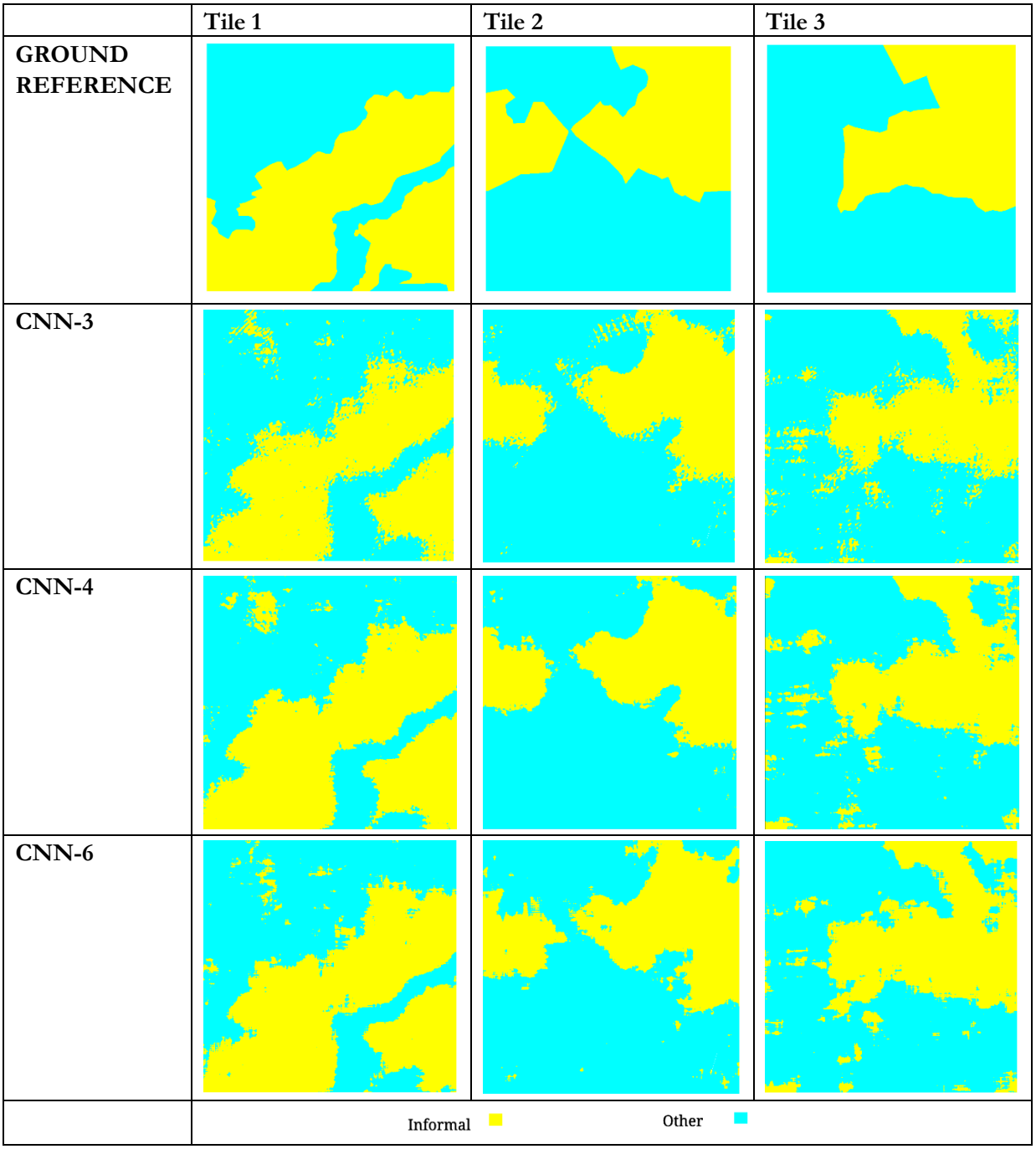


Figure D- 1: Classified maps of Tile 1, Tile 2 and Tile 3 from CNN-3, CNN-4 and CNN-6 using a training set of size 3060.

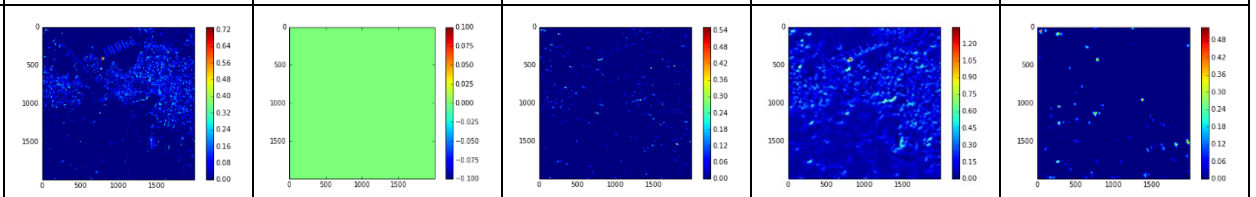
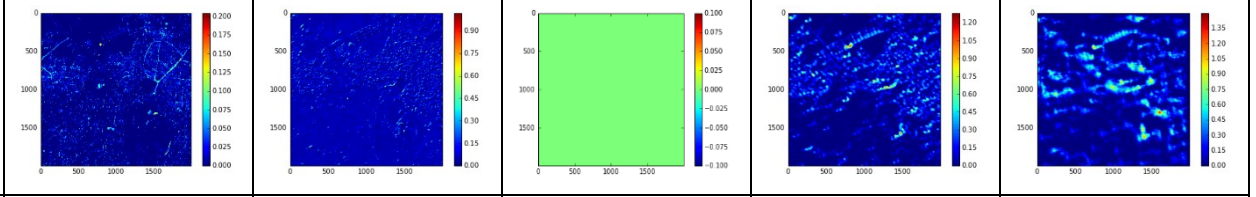
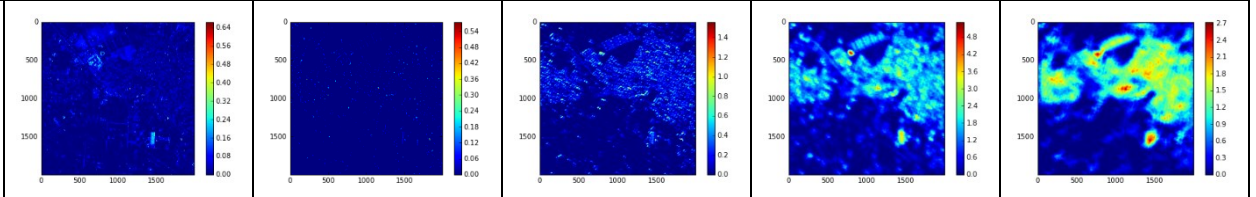
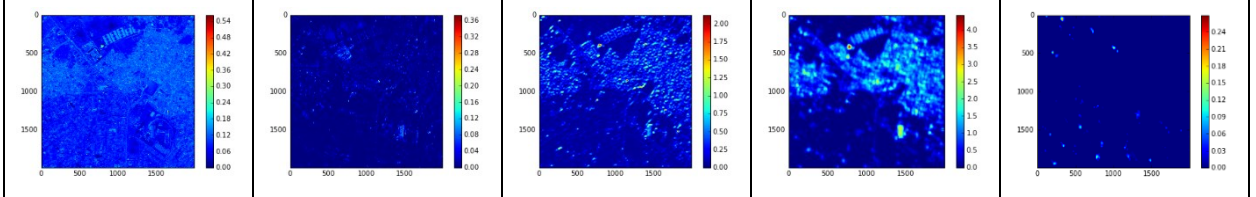
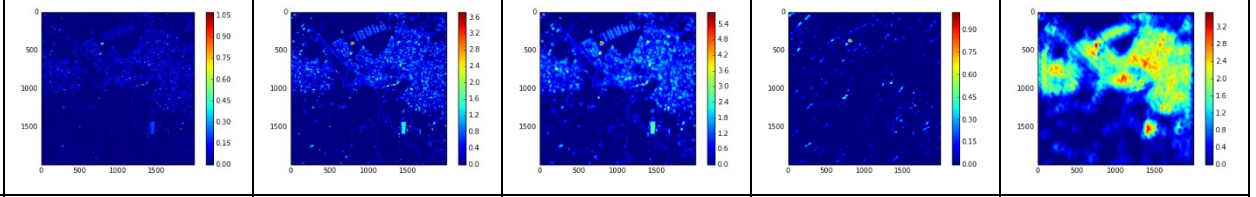
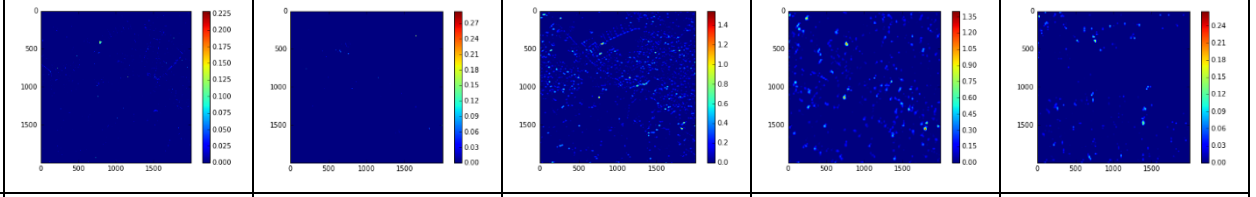
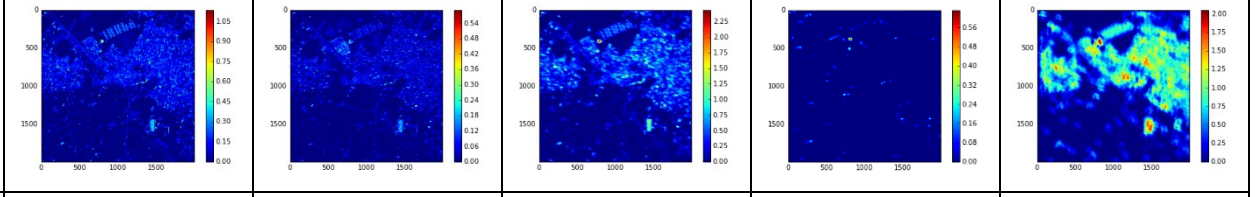
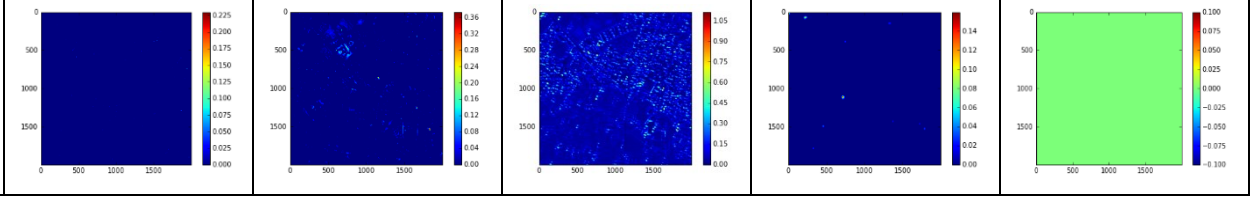
Feature Maps	Convolution layer				
	1	2	3	4	5
1					
2					
3					
4					
5					
6					
7					
8					

Figure D- 2: Feature maps for Tile 2, generated using CNN-5 trained using a sample size of 3060.

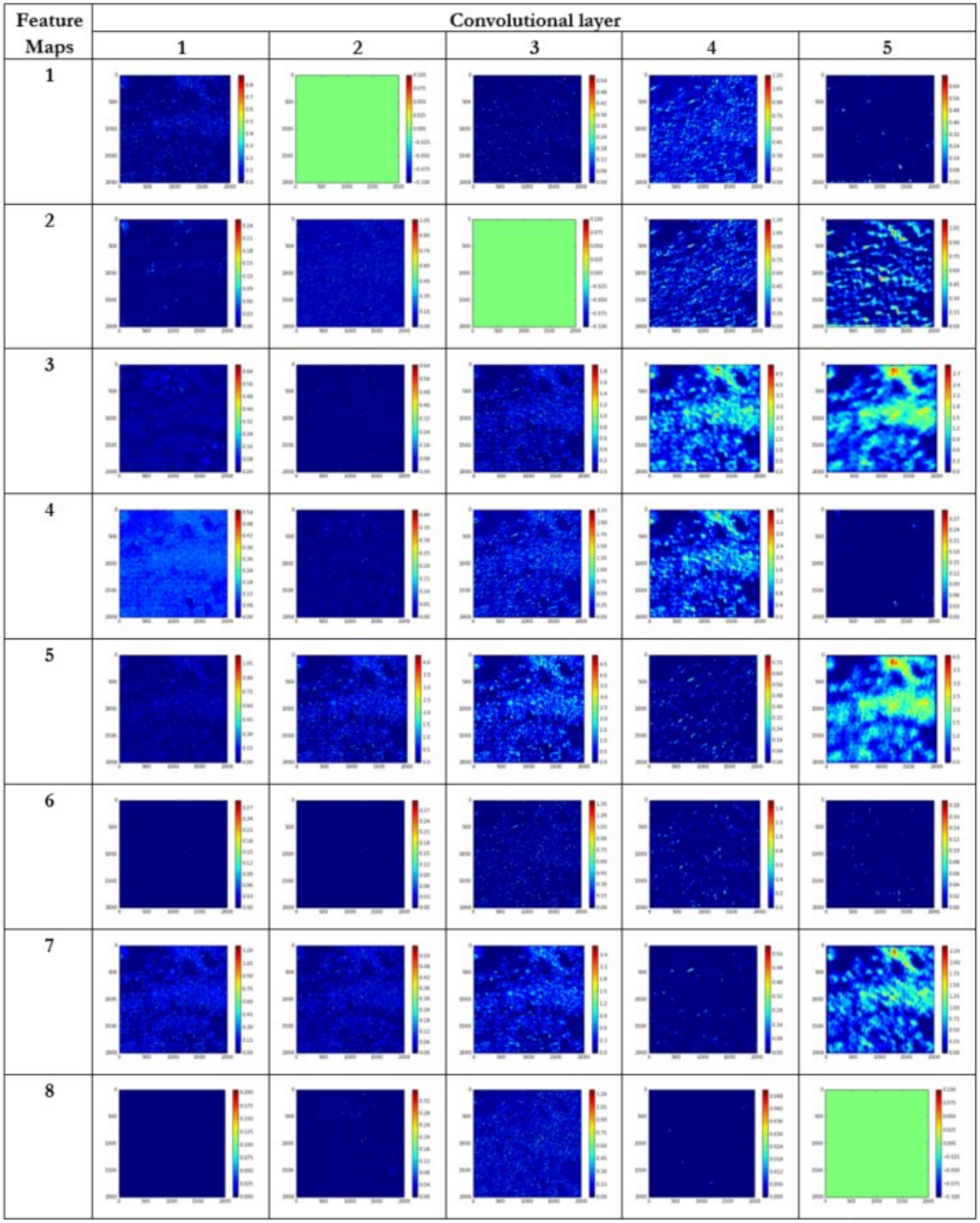


Figure D- 3: Feature maps for Tile 3, generated using CNN-5 trained using a sample size of 3060