11 August 2021

Integrated Project

Dwelling Extractions Using Deep Learning in eCognition

Sharma Pratichhya

Singh Tanya

Table of Contents

Table of Figures

Introduction

Humanitarian response requires timely and reliable information in order to function properly. It is critical to understand who, where, and how many people are affected by a specific event. Accurate numbers about the population in need, as well as the locations or areas where the affected population resides, which can be estimated using building footprint data, are critical pieces of information for almost any humanitarian intervention (Tiede et al. 2021). Deep Learning potentials have been investigated over time for automated dwelling extraction using satellite imagery to reduce the complexity caused by onsite data collection (Gella et al. n.d.). A deep learning technique that learns from the sample data fed to the model and accordingly identifies the potential dwelling matching the features of these samples can be used to extract a large number of dwellings from satellite imagery. We conducted a small study in this study for dwelling extraction using a similar deep learning technique in a new working environment, here in eCognition. A simple eCognition-based GUI (Graphical User Interface) was developed as the outcome of this study to maintain the technique's reusability and make it more understandable.

Dataset Used and Study Area

The study area in Minawao was chosen based on the available dataset. Here, WorldView imagery shown in Figure 1 was taken as test set on which training sample shown in Figure 2 as red polygons was applied on it.

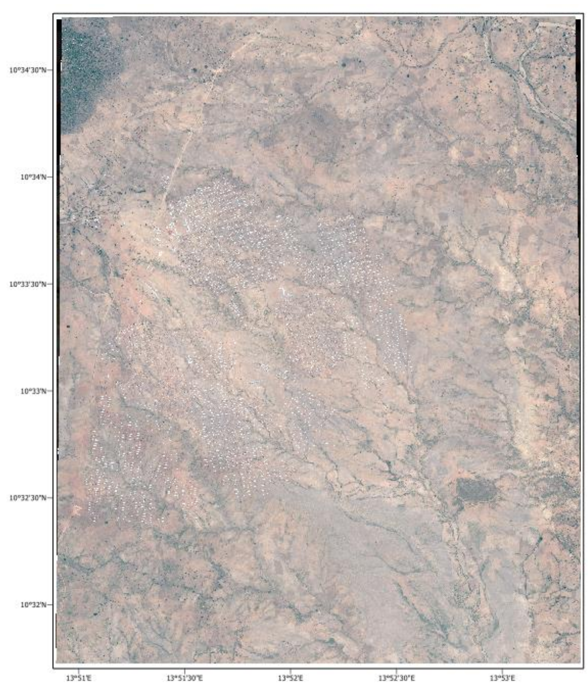

Figure 2: Satellite Imagery for study area

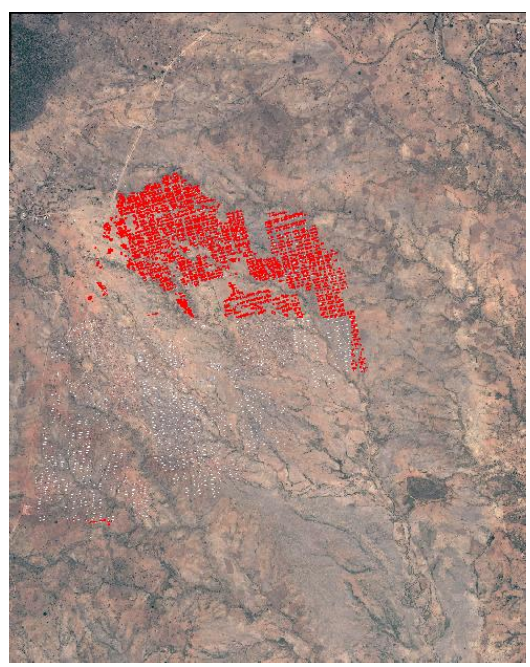

Figure 1: Sample polygons over the study area

The sample dataset provided to us included almost all dwellings, thus for testing the performance of the model and to avoid the issue of overfitting only a small portion of the entire set was taken into consideration.

Methodology

We divided the entire workflow into two major phases for developing an eCognition-based GUI that extracts dwellings from a satellite image using deep learning techniques. The first phase was dedicated to developing a ruleset, while the second phase involved the development of an architect solution for GUI.

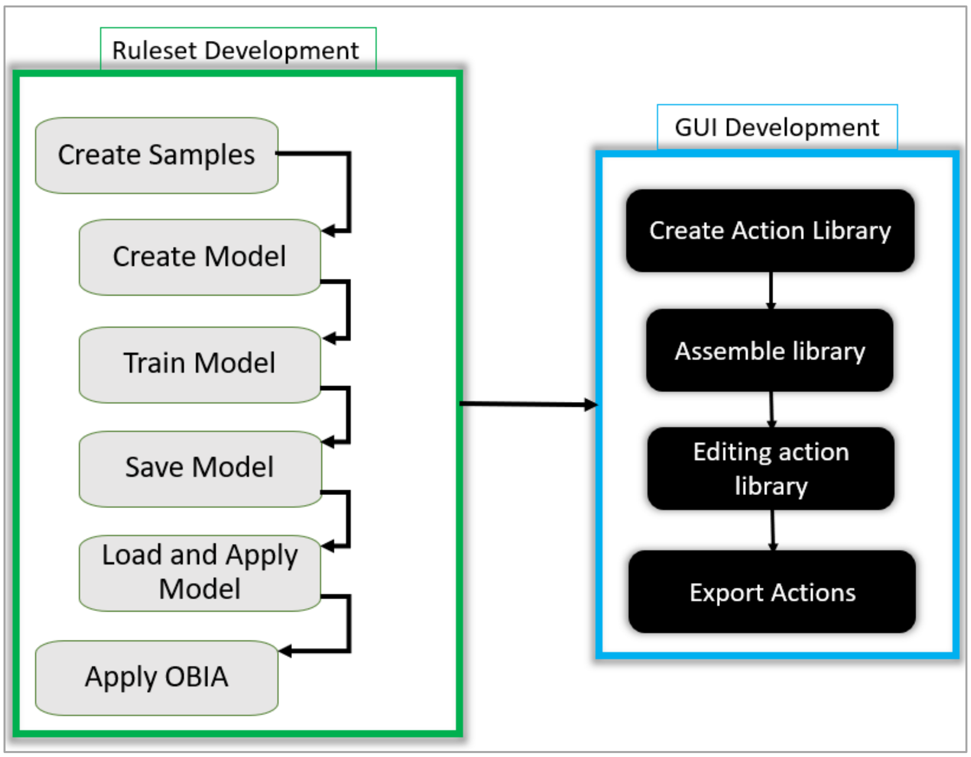

Figure 3: Study workflow

Ruleset Development

Every single process was carried out as a ruleset in eCognition. Working as a ruleset has the advantage of breaking down the image analysis method into multiple simple manageable steps, allowing the analysis to be more flexible. As shown in Figure 3, the ruleset development process includes multiple simple steps that are discussed as follows.

- Create Samples

  An image sample for CNN is called image patches in eCognition which is a square raster for a defined size and can contain information of multiple layers. This task included a method of vector-based image segmentation for generating image objects which are then classified with a specific label based on the information in the shapefile (here labelled as dwellings). The generated label sample patches as shown in the figure 4 were saved as TIF type and a sample size of 32bit.

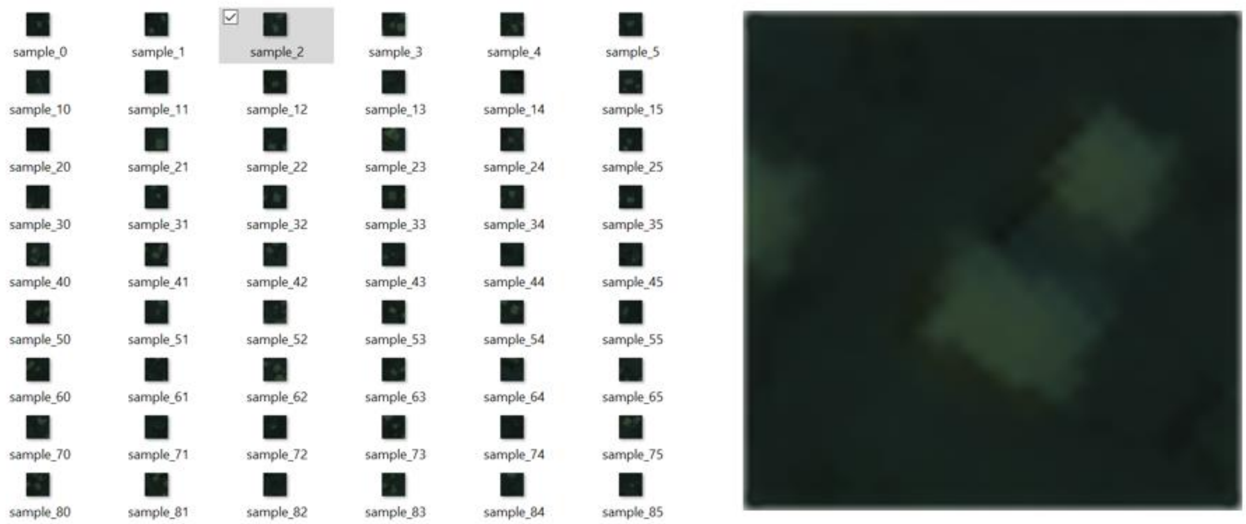

Figure 4: Generated Sample Patches

- Create Model

  Due to the assignment of several parameters, creating a model in eCognition can be a complex process. Although knowledge of these parameters (hidden layers, kernel size, max pooling, etc.) is required, we used the default settings in this study to simplify the process. In the ruleset, an algorithm for creating a convolution neural network was used, as shown in figure 5, with two hidden layers, a kernel size of three, and avoidance of max-pooling in this case.

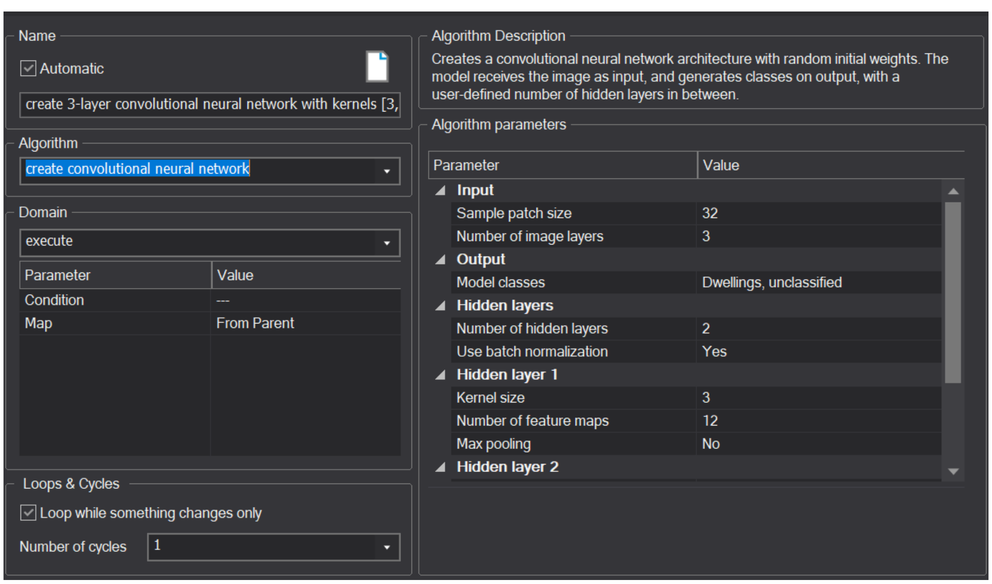

Figure 5: Create CNN model

- Train Model

  Similarly, knowledge of the learning rate, training steps, and batch size was required when training the model. A ruleset for shuffling samples was included in the training step to create a more robust model that would not be limited to a fixed order for sample selection. The train convolution neural network algorithm was then used with its default settings. Figure 6 shows the training model ruleset setup.

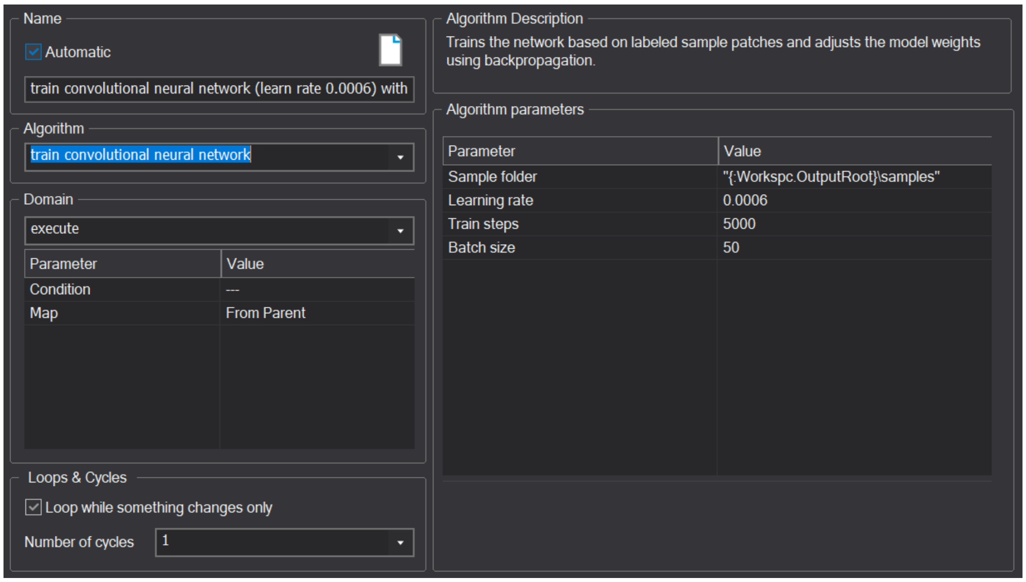

Figure 6: Train CNN model

- Save Model

  With the algorithm "Save convolution neural network" set to its default settings, a new process was created. The purpose of saving the model is to make it reusable in another scene if needed. For this study, however, we will apply the trained model to the entire scene.

- Load and Apply model

  After loading the model with the available algorithm "load convolution neural network," we can apply the model using "Apply convolution neural network," which results in a raster layer. In this case, we had to consider the image layers that were used to train the model. The output layer, HM dwellings, was then created as a heat map for extracted dwellings. Once executed heat map was achieved as the product of the CNN workflow. The image object was then created for further classification using a multi-resolution segmentation process and then classified using expert knowledge.

- Apply OBIA

  Expert knowledge is used in this step to extract dwellings from the CNN, making the outcome extremely valuable because we are combining deep learning functionalities with object-based image analysis. The result achieved with the execution of the process involved in this step can be then exported as shapefile to be used in decision making by end-users.

GUI Development

Blocks of previously created rulesets are combined as an action in a library for the development of a reusable application for different datasets. These configured actions can perform various tasks and represent a ready-to-use solution for image analysis tasks. An action library is a collection of action definitions that allow users to specify actions and put together solutions. We packaged rule set pieces in the Analysis Builder window, each of which solved a specific part of a solution. In addition, we created various user interface components (called widgets) for an action library user to adjust action parameters.

- Create Action Library

  Before using the ruleset in action definition, we created a new action library named as dwelling_action library from Architect menu in the menu bar. This library was then loaded to the Analysis builder window for further assembling and editing.

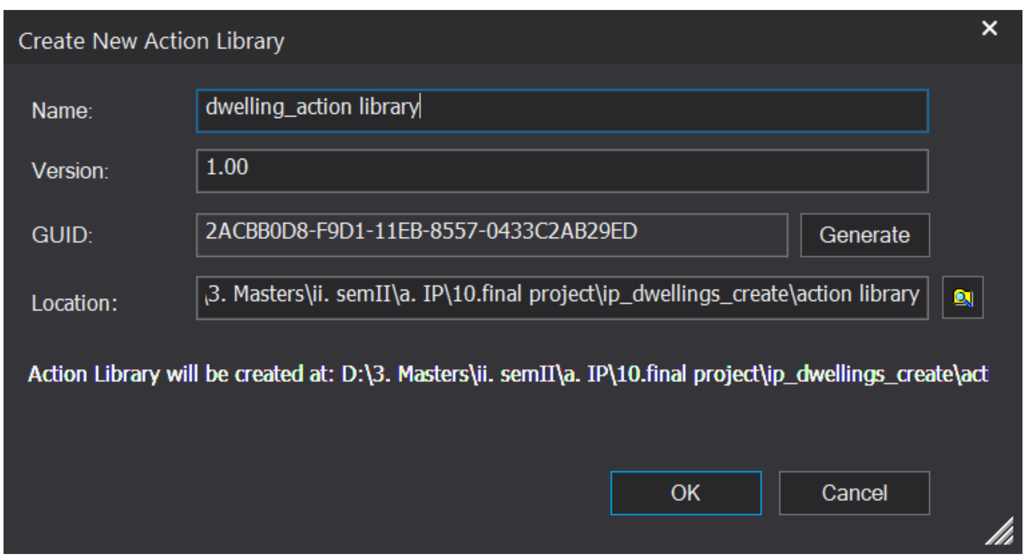

Figure 7: Creating a new action library

- Assemble Library

  Now for assembling this newly created library, we wrapped ruleset that were previously created as action definitions and gave them a user interface. This task was done for each ruleset. Depending on the requirements of the parameter we could add different features to these actions. Also, here we could add, edit or delete general settings definitions; a grouping of actions; and adding widgets for setting up properties of actions. It should be noted that the changes on the action library could only be done in editing mode which could be activated from the Architect menu.

- Editing Action Library

  Editing of Action Library includes editing of action groups, editing of action definitions or editing of action dependencies.

  Groups hold up the actions and we can create a new group if an action is not appropriate for the existing group. Each time for creating a new group we simple right-clicked on the action library background and choose Add group. Here, we could give a name to it, edit the background, text and shading colour. Also, it was easy to move the action group up or down by dragging it.

  Action definitions are unconfigured actions, which enable users of action libraries to specify actions that act as building blocks of a specific solution (Trimble 2019). For editing our action definitions, we used the ruleset prepared in the ruleset development phase. In the Add Action Definition dialogue box, we focused on adding the ruleset file, process to execute, name and its description. The process was repeated for all the actions needed for the extraction of dwellings.

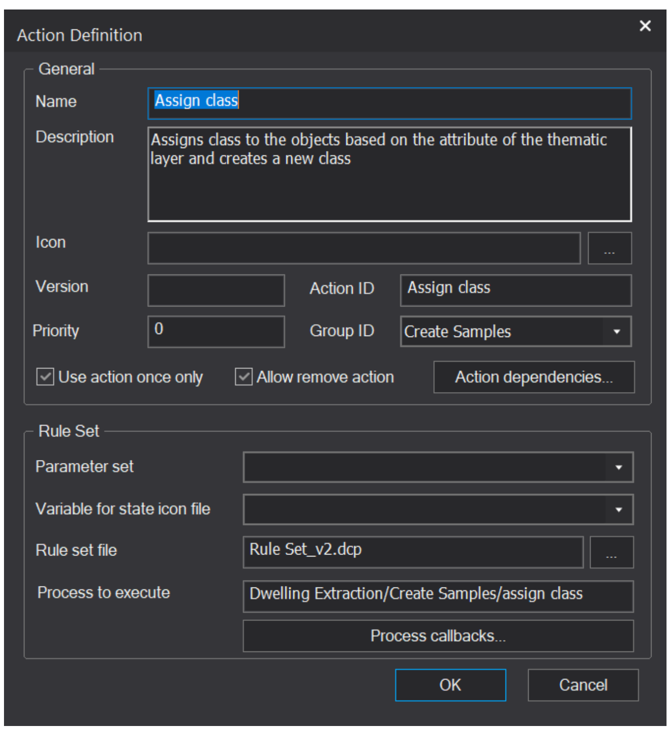

Figure 8: Editing Action Definition

- Export Actions

  The created action library was exported from Architect menu and export action so that the library can be shared and used by different users.

Result

We extracted approximately 14,000 dwellings from the deep learning technique in eCognition as a result of the entire workflow. Figure 9 depicts the studied area and the extracted dwellings. A few hundred previously digitized training samples were used to achieve the result. As shown in Figure 10, an eCognition-based deep learning technique was used in the ruleset development phase to achieve this result. Finally, to make the technique reusable, a graphical user interface (GUI) is created, as shown in Figure 11, where the user can simply click on execute to process several steps of the extraction procedure. *The shapefile, ruleset and the action library are submitted with this document.*

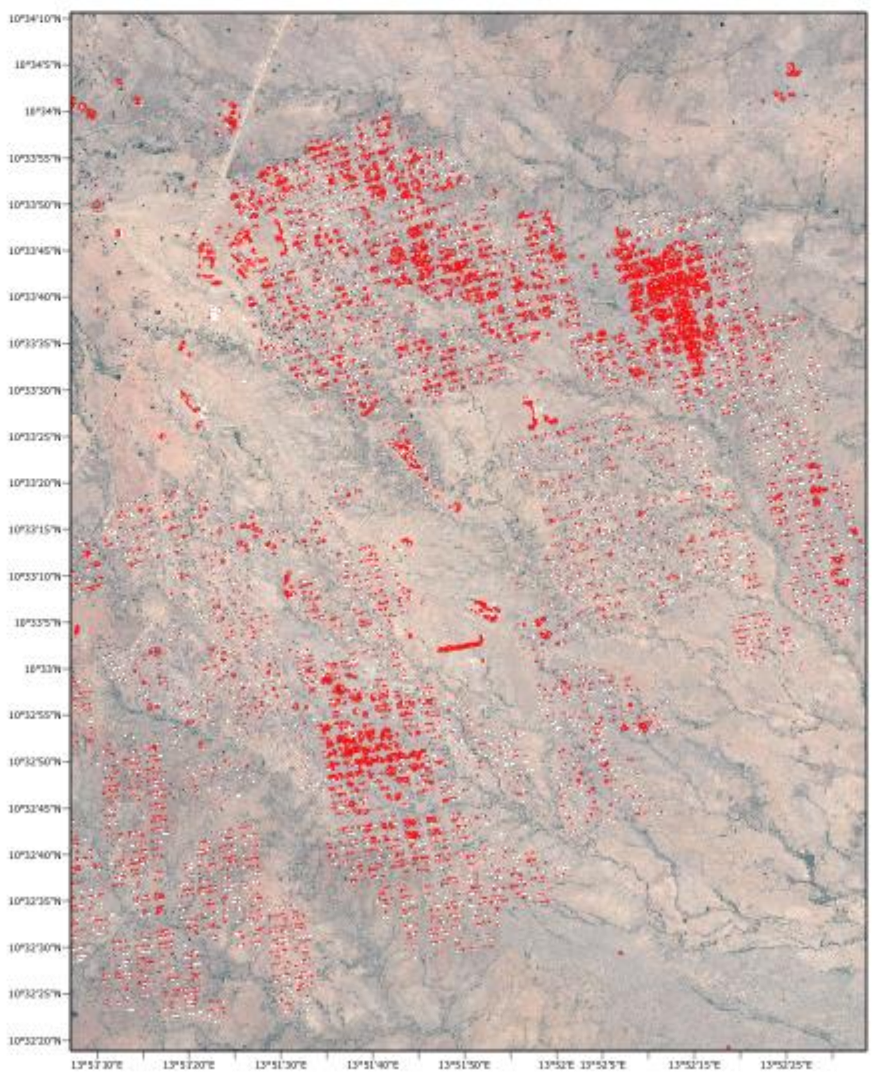

Figure 9: Dwellings Extracted

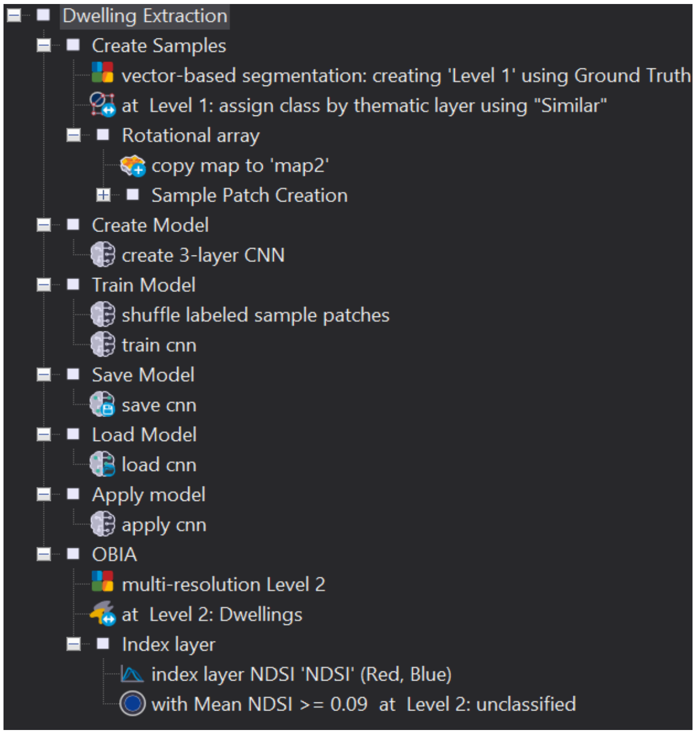

Figure 11: Ruleset Developed

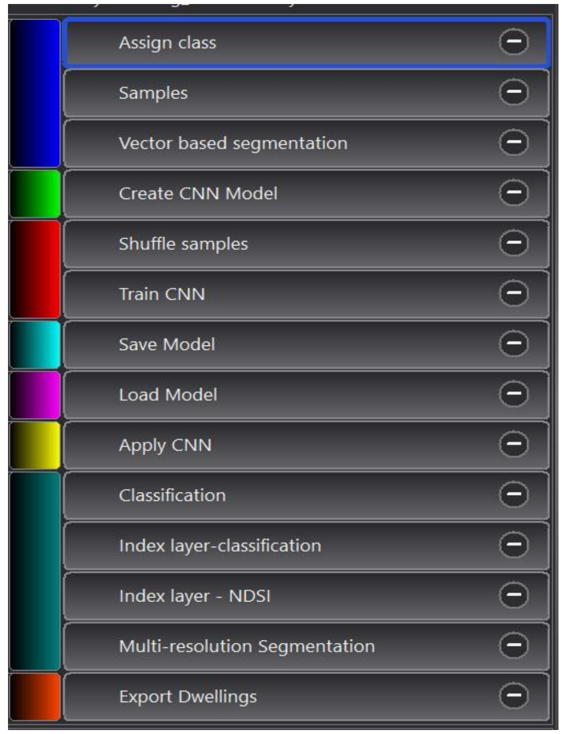

Figure 10: Action library Developed

Though numerical accuracy was not computed in this process, the model performed well in detecting the dwellings visually. We assume that our result is not very accurate because our goal was to learn the technique rather than produce a final product, but we believe that the ruleset we developed is simple and can be reused or modified for future development.

Conclusion and Discussion

Our workflow for the eCognition-based Deep Learning technique demonstrated good results for dwelling extraction using the VHR dataset. Though the result obtained is satisfactory, we can improve the method by changing the portion of the training set to avoid misclassified dwellings. Furthermore, our ruleset lacks a detailed expert knowledge concept, which necessitates additional research on the subject. Future work could fixate on reusing the model and improving it.

References

Gella, Getachew Workineh, Lorenz Wendt, Stefan Lang, Andreas Braun, Dirk Tiede, Barbara Hofer, Yunya Gao, Barbara Riedler, Ahmad Alobaidi, and Gina Marciela Schwendemann. n.d. "Gella et Al Testing Transferability of Deep-Learning-Based Dwelling Extraction in Refugee Camps." doi: 10.1553/giscience2021_01_s220.

Tiede, Dirk, Gina Schwendemann, Ahmad Alobaidi, Lorenz Wendt, and Stefan Lang. 2021. "Mask R-CNN-Based Building Extraction from VHR Satellite Data in Operational Humanitarian Action: An Example Related to Covid-19 Response in Khartoum, Sudan." *Transactions in GIS* 25(3):1213–27. doi: 10.1111/tgis.12766.