CS230

Project Sunroof

Pranjal Patil, Vedang Vadalkar
Department of Civil and Environmental Engineering
Stanford University
ppatil@stanford.edu, vedangv@stanford.edu

Abstract

Solar energy has become a vital part of the energy infrastructure and attempts are being made to achieve net zero energy buildings in California by 2020. Easy prediction of the solar energy potential of rooftops will lead to easy adoption of the solar energy systems. In this project we apply deep learning networks to aerial imagery to semantically classify roof and non-roof pixels. We use convolutional neural network based on U-Net as a starting point for our project. Each of these models were found to have high accuracies and comparable performances to current state of the art models.

1. Introduction

1.1. Motivation

Solar energy has become cost effective, and more homeowners are turning to it as a possible option to decrease their electricity bill. Having a simple but efficient tool to estimate the amount of savings will help the users to make informed decisions regarding solar energy for their residential needs. This type of tool will also aide California in achieving its goal of reaching net zero energy by 2020. In future this project can be extended to developing countries where this inexpensive method of forecasting savings will lead to increase in solar energy adoption.

1.2. Outline of the model

We will apply deep learning frameworks to aerial imagery to get the maximum annual solar energy generation for all roofs in the given image. We benefit from the extensive research in image segmentation of satellite imagery for US. We used the basic frameworks available online as a starting point in our project. We used convolutional neural network architecture, based on U-Net architecture for satellite images. In this report we find that the rooftop area can be estimated with a relatively high accuracy which validates the use of these techniques in future contexts.

2. Related Work

Traditional approaches used different wavelength reflections from objects to classify them based of the bandwidth of their reflections [1]. This approach is used in conjunction with ArcGIS software and the help of machine learning algorithms such as random forests. Satisfactory levels of accuracy were obtained through these initial models. These models were also further extended to demarcate all the different features such as roads, buildings, vegetation etc.

Further we studied some papers which can be used in deep learning to semantically classify building footprints. This was as close to roof segmentation as we could have. We can cite [2], [3], [4]. The first paper is on fully convolutional network and understanding how it works. The second one is the usage of U-Net applied to segmentation of biomedical images. The task performed in this paper is quite similar to what we aim to achieve. The third paper is also an application of convolutional neural net for aerial image processing. Ideas from all these papers were adopted and used while working on the project.

3. Dataset

The dataset in the form of images was obtained from Massachusetts Building Dataset [5]. There are two types of images provided for each data point. The first is a simple three band satellite image. The second image is a mask of just the building footprint. We are assuming roof area of a building is equal to its footprint as visible in an aerial image. This assumption makes this dataset useful for our project. All the images are in TIFF image format. The image dimensions were 1500 pixels wide by 1500 pixels height. There were a total of 150 images in that dataset. This number sounds low for a deep learning project but since the dimension of each image is much higher we were able to augment the data in order to make it viable for our project.

Data Preprocessing was carried out to make the data useful for analysis. Pixel by pixel binary classification technique was to be used that is, every pixel was classified whether it is part of the roof or not. To match the dimension of the model each image had to be converted to a 224*224 image. First, we tried resizing the images. This led to loss of lot of the features and properties of the image. Finally, we decided to take different crops of the image of size 224*224. This meant we could generate 36 images out of a single image of 1500*1500. This resulted in our dataset being augmented and we now had 5400 images of size 224*224 to be fed as input into the model. Before inputting the images, they had to be converted into arrays. Same procedure was done for all the mask images as well. We split the dataset 90%/5%/5% train/dev/test ratio.

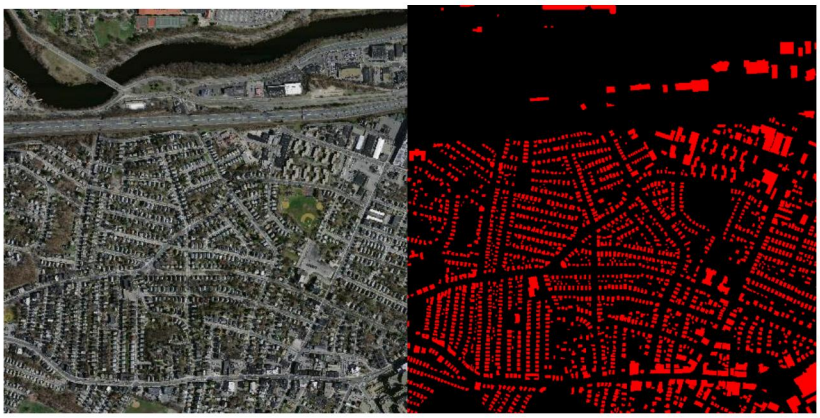
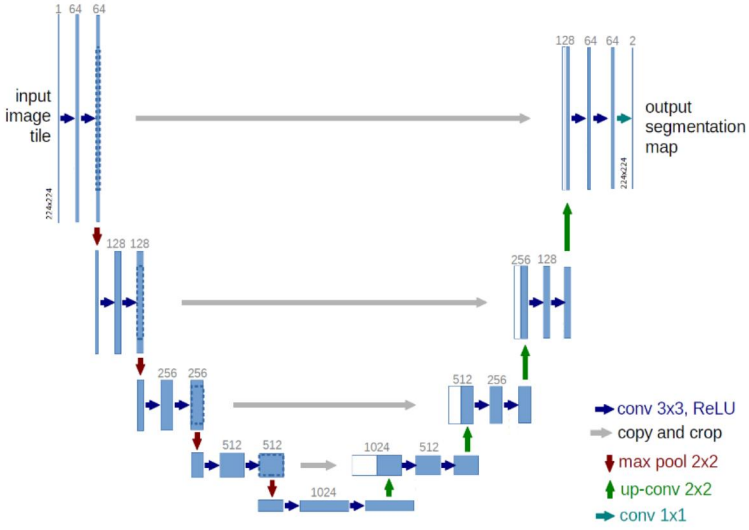


Figure 1: Input image (left), Ground truth image(right)

4. Methods

The final model developed was a based on convolutional neural network based on the UNet architecture taken from ZF-Unet [6]. This U-Net architecture was adopted from the github Repository and was further tailored and fine-tuned to achieve the final results. In the Unet architecture a downsampling path extracts features of different levels through a sequence of convolutions, ReLU activations and max poolings. This contracts the image and allowed to capture the content of each pixel. This is followed by upsampling carried out by deconvolutions. This is done to increase the resolution of the detected features. The number of features is doubled at each level of Downsampling. In U-Net architecture skip connection between the contracting and expanding part are also added as can be seen in the figure below.



Fine tuning was carried out by us since the model adopted was created to classify biomedical images. First, we did not use the pretrained weights and perform transfer learning. Rather we trained the model from scratch and estimated all the parameters. Initially we trained it as it was to create a set a baseline performance which had to be improved to go to state of the art performance. On observing the loss curve and accuracies,

we replaced the stochastic gradient descent to Adam optimizer for the model since it is known to converge faster. Also, the loss function was changed by us to simple binary cross entropy since our classification task had just one class. Finally, to improve the performance we increased the model complexity and added another down sampling layer of depth 1024. The metric used for determining the efficiency of the model was Accuracy.

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

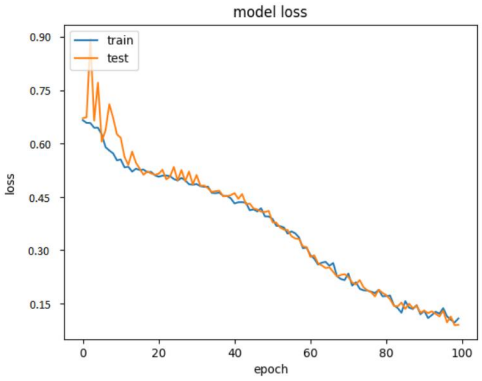
Accuracy = Total number of correctly labelled pixels/ Total number of pixels

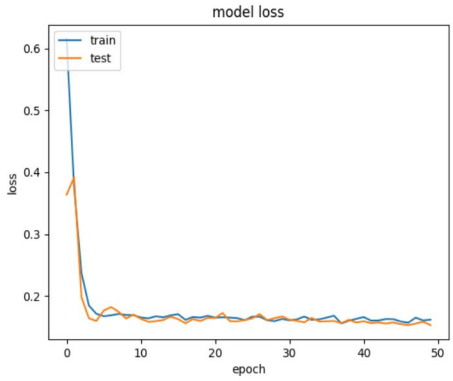
5. Experiments and Results

The training of the model was carried out on a g3.4xlarge GPU instance on AWS. The minibatch sizes chosen for the training was 16 with 50 epochs. It took over 3 hours to train the model. The learning rate was set to 0.001. There was a callback in the code by default which reduced the learning rate if the loss plateaued and remained same for more than 5 epochs.

The following table summarizes the results of the experiments carried out during the study and the figures show us the loss curve for each of the models.

Model	Validation Loss	Validation Accuracy	Training Loss	Training Accuracy
11-Layers U-Net (Max filters 1024)	0.039	0.921	0.027	0.953
9-Layers U-Net (Max filters 1024)	0.192	0.881	0.189	0.896





All the effects of the modifications in the base model can be seen in the two plots. The modifications helped achieve more accuracy with faster convergence. The final convolutional neural network performed best. Final accuracy of 0.92 was observed. There is a steep drop during the initial epochs and then plateauing was observed. No early stopping was needed as validation loss decreases throughout all the epochs for the convolutional neural network. Moreover, no overfitting was observed for the training set and that's why dropout was not adopted in the final model.

6. Discussion

We can see from the results above that we achieve an accuracy slightly inferior than state-of-the art results using convolutional neural networks as listed in the INRIA papers [7]. To try to understand the loss of accuracy of the results we calculated the precision of the final model. It came out to be 0.83. This means that the model failed to identify roof pixels with the accuracy it could identify non-roof pixels. We doubt that this has happened due to class imbalance in our input data. We tried to avoid it by thresholding and only taking images having more than 20% roof pixels, but this decreased our dataset size to an extent that we would not have been able to train any model on it. Therefore, this remains as one of the drawbacks of the model.

7. Conclusion/Future Work

In this project we tackled the problem of semantic segmentation of aerial Images to identify roofs. We used U-Net architecture with the Massachusetts Building Dataset to build our project. We found that our model prediction were less than the state of the art models in the field. We also found drawbacks like class imbalance in our data which we could not resolve due to limited time and would be our focus moving forward with the project. Building upon this work we would like to go to the next steps of the project which include adding further analysis after detecting the roof pixels that will help in detecting the size of the roof and the orientation of the roof. Also, we have to start differentiating between the roofs with solar panels vs those without any solar panels. Lastly more data collection of similar kind worldwide would help implement this model even in developing countries.

8. References

1. https://github.com/amandaJayanetti/BuiltUpAreaExtraction
2. J. Long, E. Shelhamer and T. Darrell, *Fully Convolutional Networks for Semantic Segmentation*, University of Berkeley, Proceedings of the IEEE, 2015
3. O. Ronneberger, P. Fischer, T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany, 2015.
4. Jiangye Yuan: *Automatic Building Extraction in Aerial Scenes using Convolutional Networks*
5. Masachussets building dataset: https://www.cs.toronto.edu/~vmnih/data/
6. ZFTurbo, https://github.com/ZFTurbo
7. INRIA Aerial Image Labeling Dataset, https://project.inria.fr/aerialimagelabeling/

Contributions

During the course of the project we mainly discussed about data acquisition, choice of architecture, implementation, analysis and conclusion. Pranjal worked on data acquisition, preprocessing and finding models online and implementing in Python. Vedang contributed to running the model on AWS, analysis and generating plots etc.