BUILDING ENVELOPE OBJECT DETECTION USING YOLO MODELS

Norhan Bayomi,
Mohanned El Kholy,
John E. Fernandez

Massachusetts Institute of Technology
77 Massachusetts Avenue - Cambridge, MA, USA
{nourhan,mohanned,fernande}@mit.edu

Senem Velipasalar

Syracuse University
Syracuse, NY, USA
svelipas@syr.edu

Tarek Rakha

Georgia Institute of Technology
Atlanta, GA, USA
rakha@design.gatech.edu

ABSTRACT

Building performance significantly influences energy use and indoor thermal conditions tied to the quality of living for its occupants. Therefore, information on building envelopes is essential, especially considering that envelopes and windows can impact 50% of energy loads in the United States. However, current retrofits supporting Building Energy Modelling (BEM) tools face multiple barriers, including time consumption and labor intensity due to manual modeling and calibration processes. This paper proposes using Deep Learning (DL) -based object detection algorithms to detect building envelope components, more specificlly doors, and windows, that can be applied to building energy performance analysis, 3D modeling, and assessment of thermal irregularities. We compare four different versions of the state-of-the-art YOLO V5 model to identify which version best suits the goal of detecting these building components. Results show that YOLO V5_X provides the best performance for detection accuracy.

Keywords: object detection, deep learning, BEM, artificial neural networks, building performance

1 INTRODUCTION

Advances in machine learning and computer vision algorithms have enabled the utilization of these models in 3D reconstruction, urban planning, and building restoration. In the field of building performance assessment and energy modeling, information on building envelopes is essential, where envelopes and windows can impact over 50% of energy loads in the United States (U.S. Department of Energy 2014). Thus, comprehensive and accurate building envelope audits are essential to maximize building energy savings realized from envelope retrofits. However, current retrofits supporting Building Energy Modelling (BEM) tools face multiple barriers, including time consumption and labor intensity due to manual modeling and calibration processes. In addition, identification of building envelope defects is primarily conducted using manpower, a time-consuming process that can lead to casualties and potentially life-threatening conditions.

Unmanned Aerial Vehicles (UAVs)-based photogrammetry coupled with thermography has been proven to be efficient in data collection for building thermal envelope performance (Rakha et al. 2018), 3D mapping, and acquisition of high-resolution images for building façade monitoring (Haala and Kada 2010). Furthermore, the advances in computation approaches such as Artificial Neural Networks (ANN), particularly Deep Learning (DL), have resulted in significant performance improvement in various applications, such as object detection and image segmentation (Boonpook et al. 2018). In recent years, UAV technology and DL methods have been applied for many applications, such as location identification, vehicle detection (Ammour et al. 2017), and the classification of UAV images at a much higher accuracy (Liu and Abd-Elrahman 2018). This paper deals with autonomously detecting building envelope components, namely, doors and windows, from UAV data, using a Deep Learning (DL)-based approach. The proposed approach extends our previous research that assesses building envelope thermal performance using aerial thermography data (Bayomi et al., 2021). In this work, we use a deep neural network architecture to detect envelope doors and windows and classify different thermal anomalies detected in thermal imaging data, expanding the work of (Kakillioglu, Velipasalar, and Rakha 2018).

Object detection has been gaining increased interest due to its wide range of applications, such as building façade parsing, autonomous vehicles, drone image analysis, robotics development, and transportation surveillance (Jiao et al., 2019). Numerous computer vision methods have been examined for building façade parsing (Liu et al. 2020) and façade 3D reconstruction (Hu et al. 2020). For the detection of envelope doors and windows, Sun et al. have proposed an approach for window detection through surveillance videos to determine the windows opening based on the luminance factor in image data (Sun, Lin, and Li, 2021). The method relies on standard image processing using window features and luminance to detect window openings. With the advances in Artificial Intelligence (AI), DL-based detection algorithms have been widely applied to extract complex features with high detection accuracy. Object detection approaches using DL can be classified into two main categories (Jiao et al. 2019): i) One-stage detection algorithms such as the You Only Look Once (YOLO) series (Redmon et al. 2016) and Single Shot Detector (SSD) (Liu et al. 2016), and ii) two-stage detection algorithm, including Region-based Convolutional Network (R-CNN) (Girshick et al. 2014) and fast R-CNN (Girshick 2015).

This paper uses the YOLO algorithm to detect doors and windows in RGB data collected from UAVs. We compare the performance of multiple versions of YOLO v5, a state-of-the-art object detector, to predict the locations of windows and doors. The following section presents an overview of previous applications of numerous DL approaches in object detection, emphasizing the architecture and development of YOLO algorithms and their application in object detection. YOLO model is specifically chosen as it has a very close accuracy to *EfficientDet* (which is known to surpass other object detection algorithms like Faster RCNN and Single Shot Detectors (SSD)), but with a much higher speed (approximately ten times faster). In addition, YOLO algorithms can be scaled up without affecting the speed as much as to SSD.

2 RELATED WORK

Object detection is the process of identifying an object in an image with its classification and localization (Khan et al., 2020). Object detection has been widely used in numerous applications, yet it's considered one of the most challenging fields in computer vision. Several studies have proposed deep learning networks as a backbone for object detection and feature extraction either from image data or videos (Alzaabi et al., 2020). There are numerous domains in object detection, such as scene text detection, face detection, multi-categories detection, and edge detection (Khan et al., 2020). With advances in computational capacity, the development of numerous Convolution Neural Networks (CNN), and deep learning (DL), object detection has progressed in speed and accuracy. DL is a family of models that consists of multiple matrices, and each matrix is called a layer. When training data is introduced to these models, the values of the matrices keep changing until the model matches the training data and its corresponding labels. This process is called learning and usually happens on multiple iterations (each iteration is a complete cycle in which the model gets exposed to every data point in the training data). The number of iterations varies significantly according to the model structure and the training data size. CNN is a proposed iteration on deep learning models in

which the model uses convolutional filters on the image in the middle layers. The convolutional layers give the model a great advantage since, for a given pixel, only the nearby pixels take attention. Thus it saves the model unnecessary operations with irrelevant pixels. Consequently, the model will do fewer mathematical operations and become faster.

The first object detection model was introduced in 2001, called the Viola-Jones detector; it was adopted by many computer vision libraries and primarily used for face detection (Viola and Jones 2001). The first DL-based object detector was the Overfeat Network which was introduced in 2014 using Convolution Neural Networks (CNNs) (Sermanet et al. 2014). Using CNNs in object detection has introduced new network architecture that immensely assisted in advancing the state-of-the-art.

There are two types of object detectors, a one-stage detector and a two-stage detector. The two-stage detector performs detection over two steps. The first step is called the Region Proposal Step, which generates a set of regions in an image with a high probability of containing an object. The second step is the Object Detection Step, which uses the output regions as an input to perform the final detection and classification. The one-stage detectors combine these two steps and detect an object and its class directly from an image, making them more efficient and suitable for real-time detection applications (Jiao et al., 2019).

2.1 Two-Stage Detectors

Since 2012, object detection models have witnessed significant breakthroughs with the introduction of R-CNN, which improved detection accuracy by 30% over previous models (Chahal and Dey 2018). R-CNN was first proposed by Girshick (2014), where its architecture is based on three main modules: region proposal, vector transformation, and classification. The first module generates region proposals that are independent of any category. The second module extracts feature vectors from each region proposal to classify objects in a single image. Finally, the third module is the bounding box progressor that conducts the final prediction and classification. A year later, a faster version of R-CNN was introduced, Faster R-CNN (Girshick 2015). In Faster R-CNN, features are extracted from an input image. Then, the regions of interest are passed through a pooling layer to get the size features used as an input for the bounding box regressor to complete the final classification. In the R-CNN method, trained CNNs are used to classify the object area to decide if it belongs to an object (Chen and Huang 2014), while Faster R-CNN uses a VGG16-based network for feature extraction (Shlezinger et al. 2020). An extension of Faster R-CNN is Mask R-CNN which is mainly used for segmentation problems (R-cnn et al. 2020). Mask R-CNN adds a Fully Convolutional Network (FCN) branch to predict segmentation masks for each region of interest (RoI) alongside the existing branch for classification and bounding boxes. Thus, it is considered more accurate for object detection.

2.2 One-Stage Detectors

After Faster R-CNN, Redmon et al. (2016) introduced a one-stage object detector YOLO (You Only Look Once). The main idea of YOLO is it can provide real-time detection of full images using a faster pipeline by predicting less than 100 bounding boxes per image compared to 2000 region proposals in R-CNN. Since then, there has been an incremental upgrade to the YOLO model by integrating ResNet50 (He et al. 2016) and Feature Pyramid Network (FPN) (Lin et al. 2017) till the YOLO version (3) (Redmon and Farhadi 2018). YOLO deals with object detection as a regression problem where it extracts features from an image to directly predict the probability of a class. The basic structure of the YOLO model is based on an end-to-end pipeline, where an input image is divided into an SxS grid, and each cell is used to predict the object centered in that cell. The prediction process is performed by examining the center of the semantic component in each cell. Each grid produces a B bounding box with a confidence score of χ (Figure 1).

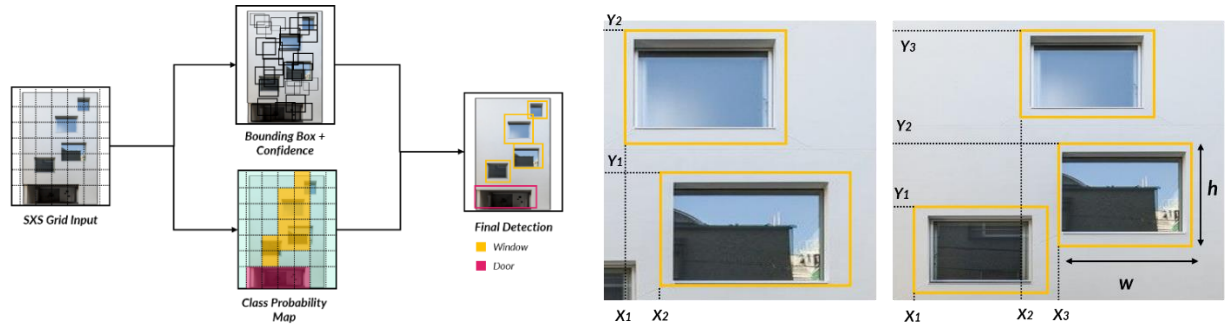

Figure 1: Left, Conceptual structure of YOLO model, Right, parameters of the model (based on (Redmon et al. 2016).

The confidence score of each predicted class is calculated using the following equation:

$$\chi = P_{class\ i} \times Q_{class\ i} \tag{1}$$

P is the probability of detected object in a bounding box B with an accuracy score Q to account for the fitness between the predicted box and target object. There is an N bounding box for every image, and each bounding box is defined by four parameters (Figure 2). The dimensions of the bounding box are determined by w, h, and x,y represents the coordinates of the upper left corner of each bounding box. After the basic YOLO version, YOLO v2 and YOLO v3 came out. YOLO v3 uses three-scale feature maps to predict the bounding box and provides a more robust feature extractor using Darknet-53 inspired by ResNet (Redmon and Farhadi 2018). Another one-stage detector is Deconvolutional Single Shot Detector (DSSD) (Fu et al. 2017) which uses ResNet-101 as the backbone and adds additional prediction and deconvolution modules to the detection pipeline. The primary purpose of the deconvolution module is to increase feature maps' resolution, where a prediction module follows each deconvolution module to enable the prediction of objects with varying sizes.

2.3 Multi-level Detector

In addition to one-stage and two-stage detectors, numerous studies have developed detection algorithms that use multi-layer representation methods to achieve more accurate feature expression (Gao, Wen, and Liu, 2017; Hou et al., 2019; Li and Yu, 2016; Shen et al. 2018). One example is ResNet-50, where a three-layer deep convolution feature is used for small object detection (Ren, Zhu, and Xiao 2018). Also, Feature Pyramid Network (FPN) proposed in (Lin et al. 2017) used a multi-scale feature fusion method to improve the performance of small object detections. The main advantage of the multi-scale is that it can provide more semantic information that improves overall detection accuracy, especially for small objects (Kong et al. 2016).

In this paper, we chose to look into a one-stage detector to deal with doors and windows detection in facades' image data. The geometrical features of doors and windows are relatively simple; however, the data size can be large, leading to more computational time. Thus, we will examine the performance of multiple one-stage YOLO v5 detectors in detecting building envelope components for accuracy, precision, and speed using data collected from UAVs and handheld cameras.

3 RESEARCH METHODOLOGY

Numerous research studies have focused on extracting and segmenting buildings' envelopes using photogrammetry and computer vision techniques. In the field of detecting building envelope objects from images, numerous models have been developed using deep learning techniques such as Recurrent Neural Networks (RNN) (Graves et al. 2008) and Convolutional Neural Networks (CNN) (Krizhevsky, Sutskever, and Hinton 2012). These models have been widely used due to their detection accuracy and assisted in

numerous fields, such as object detection (Girshick et al. 2016) and image clustering and classification (Chan et al. 2015), yet their detection rate has been slow. The trade-off between speed and accuracy is still an open problem: a more complex model with higher accuracy will be slower. Thus, in this research, multiple versions of YOLO V5 models with different sizes are compared in terms of accuracy (MAP: mean average precision) and speed (FPS: frames per second).

3.1 Model Architecture

In YOLO V5 (Zhu et al. 2021), the detection of small objects has been improved significantly, making it suitable for detecting semantic objects with repeating structures such as windows and doors in building facades. Hence, we adopt the YOLO v5 model as the main algorithm for object detection from RGB and IR images captured from UAVs or handheld thermal cameras. The architecture of the YOLO model is a convolutional neural network that consists of a backbone: CSPDarknet, Neck: PANet, and Head: Yolo layer, as shown in Figure 2. Also, YOLO V5 can process images in real-time at 78 FPS with fewer false positives in the background (Redmon and Farhadi 2018). Furthermore, since doors and windows are considered semantic objects with varying sizes and poses, YOLO V5 is most suitable for overcoming this problem by incorporating multi-scale fusion (Lin et al. 2017) to detect objects with good adaptability to changes in object sizes.

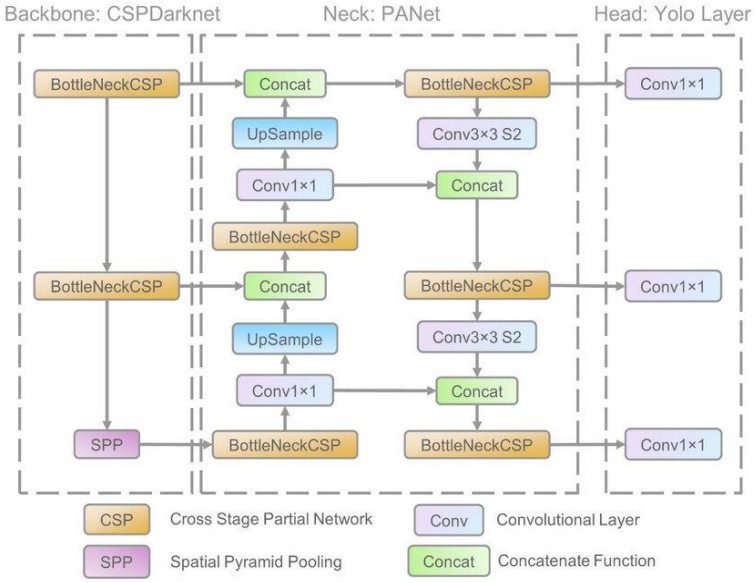

Figure 2: Network architecture of YOLO model.

3.2 CycleGAN

Since IR image annotation is not readily available in the literature, we propose to use the CycleGAN model to improve the detection of building envelope objects in the IR spectrum. CycleGAN is an image-to-image translation algorithm capable of transforming the domain of an image by learning to map between an input image and an output image through a training dataset of aligned pairs (Zhu et al., 2017). The structure of the CycleGAN model is based on using two datasets as input, and through training, the model learns to transform between two domains of the datasets. In this paper, we use CycleGAN to transfer the RGB dataset to its IR replica to improve the detection capabilities in the IR spectrum. Even though it is impossible to infer the temperature from RGB, copying the same color patterns will be sufficient as a training data set for the YOLO model. The CycleGAN implemented here consists of two generators and two discriminators, as shown in Figure 3 below. Each generator consists of three convolution blocks, nine residual blocks, and three convolution transposed blocks. The discriminator consists of multiple CNN filters as it implements the patch GAN strategy.

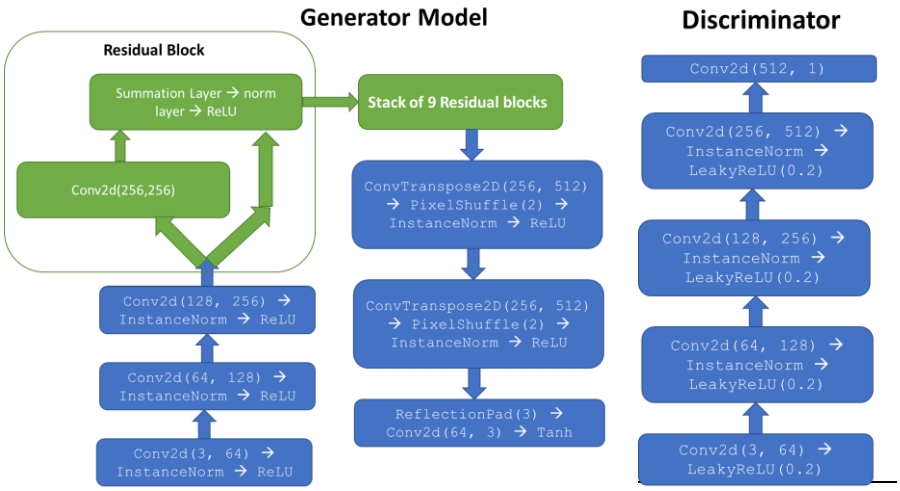

Figure 3: CycleGAN structure based on (Zhu et al. 2017).

3.3 Dataset

This paper aims to detect various building envelope components, specifically doors, and windows, in the image data of buildings using a one-stage object detection model. In the study, we selected a representative dataset of single-family and multi-family residential buildings in Boston, Massachusetts, where we defined doors and windows as the detection targets. The dataset consists of 3000 raw images split into training and testing datasets at a ratio of 98:2. In addition, we separated the validation dataset from the training dataset to fine-tune the model detection performance.

3.4 Image Acquisition

In object detection problems, the quality and quantity of the dataset are considered decisive factors for the model and thus will significantly affect the accuracy of the detection and the generalization of the model. Unfortunately, available datasets for object detection applications are limited in the context of buildings' doors and windows, especially considering the variety. In this paper, we obtained the dataset using a handheld thermal camera and a standard professional camera for higher resolution images, all captured at a perpendicular angle to the façade from the ground floor level at a distance that covers the entire façade area. Several images of building facades were collected, including images captured under different lighting conditions and obstacle occurrences (trees, cars, humans). Images covered six types of windows and four types of doors widely found in residential buildings in the U.S. We chose to capture the dataset with noise occurrences to improve the model detection performance in situations where visual obstacles may be found.

3.5 Image Pre-processing and Annotation

For the dataset collected with the handheld thermal camera, the different formats and resolutions between RGB and IR images may cause the target detection network not to read the images during training. Thus, we used high-resolution RGB images to create their IR replica using the CycleGAN model (Figure 4). Data in .jpg are retained as the standard dataset for the detection model training. All images are normalized to a standard size for model training. Next, training datasets were labeled for two categories; windows and doors, as XML files that carry the number of objects, object category, and target bounding box's four coordinates.

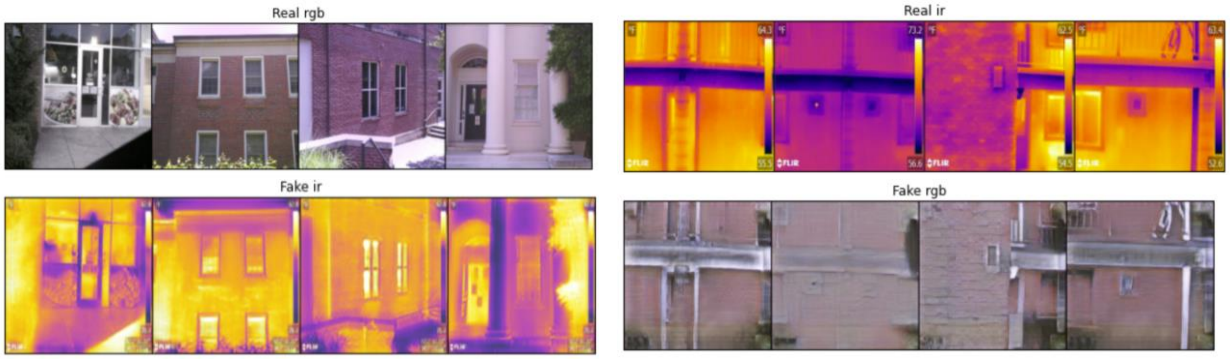

Figure 4: CycleGAN model output for RGB and IR replicas using the original collected data set.

4 RESULTS

4.1 Parameter Setting and Training Method of CycleGAN

CycleGAN is used to transform the style of the image from one format to another. This paper uses two data formats: RGB and IR. For the model to be well trained to transform a given image from one format to another (RGB to IR or vice versa), two datasets (IR image dataset and RGB image dataset) need to be provided. In our case, the RGB dataset contains 2940 RGB images, and the IR dataset contains 1989 IR images. The training was carried out in batches of images; each batch contained five images.

The CycleGAN model consists of two generators for each image format (G_{IR}, G_{RGB}) and two discriminators (D_{IR}, D_{RGB}). The G_{IR} and G_{RGB} correspond to the generators that generate IR images and RGB images, respectively. D_{IR} and D_{RGB} represent the discriminators for $G_{IR\ and}$ G_{RGB}, respectively. In this paper, the objective function is defined as:

$$L_{obj}(G_{RGB}, G_{IR}, D_{RGB}, D_{IR}) = \lambda L_{Cyc}(G_{RGB}, G_{IR}) + L_{GAN}\left(G_{RGB}, D_{Y_{IR}}, X_{RGB}, Y_{IR}\right) + L_{GAN}\left(G_{IR}, D_{Y_{RGB}}, X_{IR}, Y_{RGB}\right) \quad (2)$$

where λ=10.

The loss function of the CycleGAN is composed of cycle loss and adversarial loss. For a given generator that converts X → Y and a given discriminator, the adversarial loss is calculated using the following equation:

$$L_{GAN}(G, D_Y, X, Y) = E_{y\sim p_{data}(y)}(log(D_Y(y)) + E_{x\sim p_{data}(x)}(log(1 - D_Y(G(x))) \quad (3)$$

Given two generators, G and F, the cycle loss is defined as:

$$L_{Cyc}(G, F) = E_{y\sim p_{data}(y)}(|G(F(y)) - y|) + E_{x\sim p_{data}(x)}(|F(G(x)) - x|) \quad (4)$$

Regarding the optimizer parameters, the learning rate was set to 0.0002, and the betas were set to 0.5 and 0.999, respectively. The value of losses is reported for each 50-gradient update. Figure 5 represents the cycle loss across the gradient updates of each generative model.

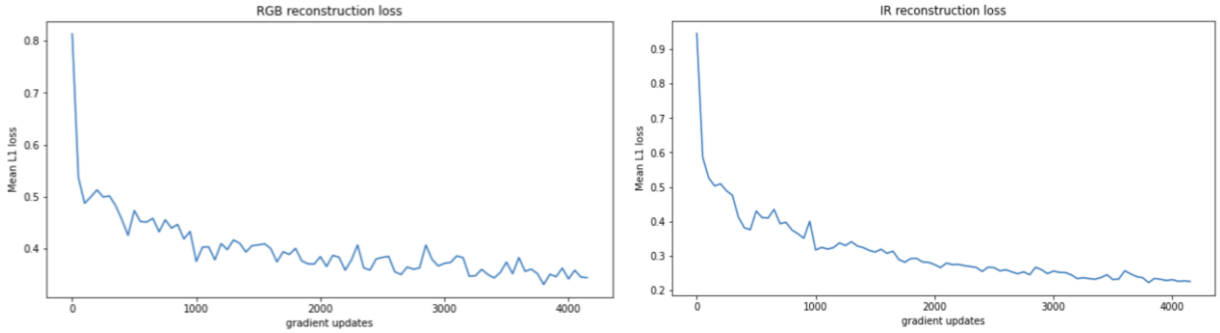

Figure 5: The cycle loss across the gradient updates in the CycleGAN model.

4.2 YOLO Model Performance

The YOLO V5 model was built in PyTorch, and we divided the data into 98% (2940 images) for training and 2% (60 images) for testing. To assess how precise the output is, we adopted the assessment method in (Hu et al. 2020b; Rahmani and Mayer 2018), where we account for every classified pixel as either False Positive (FP) or True Positive (TP), and the precision equals to TP/(TP+FP)The total calculated precision was 0.93. To assess the precision of the object detection, we tested the model with different resolutions and layout configurations, and the model performed well with low-resolution images captured by the FLIR camera. Figure 6 shows the results of the detected doors and windows using the testing data set. The orange frame represents the detected windows, and the red frame denotes detected doors. During the training process, we optimized the loss function using the following equation:

$$\lambda_{coord}\ \sum_{i=0}^{S^2}\sum_{j=0}^{B} d_{ij}^{obj}\ [(x_i-\hat{x_i})^2+(y_i-\hat{y_i})^2]+$$
$$\lambda_{coord}\ \sum_{i=0}^{S^2}\sum_{j=0}^{B} d_{ij}^{obj}\left[(\sqrt{w_i}-\sqrt{\hat{w_i}})^2+(\sqrt{h_i}-\sqrt{\hat{h_i}})^2\right]+\lambda_{coord}\ \sum_{i=0}^{S^2}\sum_{j=0}^{B} d_{ij}^{noobj}\ [(c_i-\hat{c_i})^2+$$
$$\sum_{i=0}^{S^2} d_i^{obj}\ \sum_{c\ \epsilon\ classes}(p_i(c)-\widehat{p_i(c)})^2 \qquad (5)$$

Where, in a given cell i, the coordinates of the center of the bounding box B are denoted as (x_i, y_i) to the bounds of the grid cell with normalized width w_i and height h_i relative to the image size. d_i^{obj} represents the existence of an object, c_i is the confidence of detection and d_{ij}^{obj} specifies that the jth bounding box performed prediction. The loss function penalizes classification errors only if an object is located in that grid cell i. Next, we assign a binary variable ϵ [0,1] to represent the state of the selected attributes in each bounding box.

Figure 6: Door and window detection results obtained on our testing data set.

4.3 Comparative Models and Performance Assessment

In this paper, we compared four different versions of YOLO V5 that vary in size and performance to identify the best performing version for the envelope object detection task, where **YOLOV5 Nano** is the smallest in size and < YOLOV5 Small < YOLOV5 Medium < **YOLOV5_X**, which is considered the most prominent version in size. Furthermore, we compare the output of each version based on two training datasets: RGB image data and IR image data, to validate the model performance for two different data formats. The primary objective function is based on three key parameters. First is the binary cross-entropy (BCE) loss, which is a classification loss defined as:

$$BCE = y_i * log(y_i) + (1 - y_i) * log(1 - y_i) \quad (6)$$

Second, the box validation (L1 loss) represents the mean absolute difference between the dimensions of the prediction and the original box of a given component (either a door or a window). Finally, the mean Average Precision score (**mAP**) compares the ground-truth bounding box to the detected box. We set the mAP score at confidence 50% for a given prediction. Table 1 compares the performance of the different YOLO V5 versions trained on the IR dataset. YOLO V5 model has multiple versions; each version varies in size from the others. From the analysis, we found that YOLO V5_X is the best performing version for how tight the predictions are. Also, YOLO V5_X is the model with the highest number of parameters and the slowest inference speed.

Table 1: Performance of different YOLO V5 versions on the IR data.

Training on the IR data	Best validation box loss (L1)	Best validation classification loss (BCE)	Best mAP at 0.5 confidence	Inference speed on V100 b32	GPU Memory on batch = 5	Number of parameters (in Millions)
Nano model	0.03	0.0056	0.846	**0.6 ms**	**1.9 Gb**	**1.766**
Small model	0.029	**0.0046**	**0.863**	0.9 ms	3.48 Gb	7.025
Medium model	0.027	0.00501	0.843	1.7 ms	6.25 Gb	20.87
X model	**0.026**	0.00505	0.851	4.8 ms	14.3 Gb	86.22

Table 2 compares the performance of different YOLO V5 versions in prediction based on RGB image data training. It can be noted that YOLO V5_X is also the best performing model for prediction tightness. Figure 7 represents the performance of four YOLO versions across the three performance parameters (L1, BEC, mAP) for both RGB and IR datasets across each training cycle.

Table 2: Performance of different YOLO V5 versions on RGB data.

Training on the RGB data	Best validation box loss (L1)	Best validation classification loss (BCE)	Best MAP at 0.5 confidence	Inference speed on V100 b32	GPU Memory on batch = 5	Number of parameters (in Millions)
Nano model	0.0276	0.0046	0.859	**0.6 ms**	**1.9 Gb**	**1.766**
Small model	0.0273	0.0046	0.873	0.9 ms	3.48 Gb	7.025
Medium model	0.026	**0.004**	0.868	1.7 ms	6.25 Gb	20.87
X model	**0.0242**	0.0043	**0.885**	4.8 ms	14.3 Gb	86.22

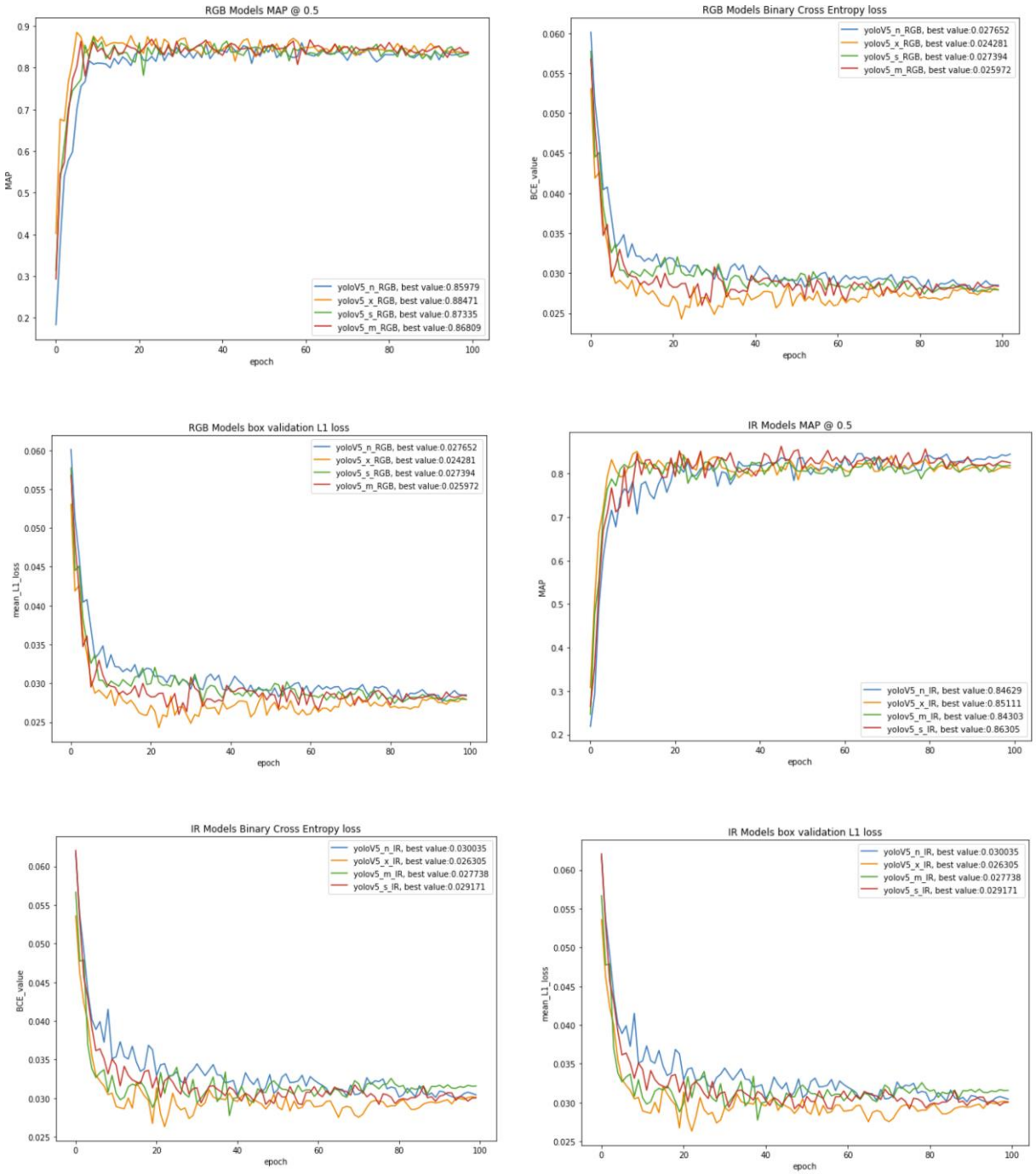

Figure 7: The performance of the four YOLO versions across training epochs.

5 DISCUSSION AND CONCLUSION

Building envelopes hold substantial energy efficiency and heat risk mitigation opportunities, especially in the residential sector. However, the lack of sufficient information on the performance of existing buildings makes it difficult to assess the impact of retrofit strategies on facing heat exposure risks. As a result, it makes tackling building envelope improvements quite challenging. This paper examines different versions of YOLO object detection models, in terms of detection performance and inference speed, to detect doors

and windows from RGB and IR image data. Providing an accurate method to detect building envelope components can inform other applications such as building envelope audit, deterioration detection, classification, and Building Energy Modelling (BEM).

More specifically, we have compared the performance of four versions of the state-of-the-art YOLO v5 model to examine which version is best suited for envelope object detection. We used real-world data collected with various image interference and noise occurrence to train the models. In addition, the dataset contained a rich collection of window and door types constructed for training and testing the model. The study showed that YOLO V5_X outperforms other versions of YOLO V5 in detection accuracy, as explained in this paper. Also, YOLO V5_X provides an improved capability to overcome obstacles in the image dataset. This was achieved by training the model using images taken across different times of the day to accommodate different lighting conditions. In addition, the advantage of YOLOV5_X compared to other models is the large number of parameters, hence more mathematical operations per image and thus more accuracy with complex images.

The work presented in this paper will help provide building energy modelers with more accurate façade information in situations with limited access to façade drawings. Also, the proposed approach of windows and door detection can be used in constructing a window-to-wall ratio that can be beneficial for building energy modeling and performance simulation. Another key advantage of the proposed approach is the classification of thermal anomalies detected in the infrared image data. For example, detected thermal anomalies close to doors or windows can have a higher probability of being considered infiltration or exfiltration. Thus, providing the capacity to detect building components autonomously will assist in thermal anomaly classification problems and automated workflow development of anomaly detection and diagnosis.

REFERENCES

Alzaabi, A., M. Talib, A. B. Nassif, A. Sajwani, and O.ar Einea. 2020. "A Systematic Literature Review on Machine Learning in Object Detection Security." *2020 IEEE 5th International Conference on Computing Communication and Automation, ICCCA 2020* 136–39.

Ammour, N., H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair. 2017. "Deep Learning Approach for Car Detection in UAV Imagery." *Remote Sensing* vol. 9(4), art. 312.

Bayomi, N., S. Nagpal, T. Rakha, and J. E. Fernandez. 2021. "Building Envelope Modeling Calibration Using Aerial Thermography." *Energy and Buildings* 233:110648.

Boonpook, Wuttichai, Y. Tan, Y.Ye, P. Torteeka, K. Torsri, and S. Dong. 2018. "A Deep Learning Approach on Building Detection from Unmanned Aerial Vehicle-Based Images in Riverbank Monitoring." *Sensors (Switzerland)* 18(11).

Chahal, K. S. and K. Dey. 2018. "A Survey of Modern Object Detection Literature Using Deep Learning."

Chan, T. H., K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. 2015. "PCANet: A Simple Deep Learning Baseline for Image Classification?" *IEEE Transactions on Image Processing* 24(12), pp. 5017–32.

Chen, B. H., and S. C. Huang. 2014. "An Advanced Moving Object Detection Algorithm for Automatic Traffic Monitoring in Real-World Limited Bandwidth Networks." *IEEE Transactions on Multimedia* 16(3), pp. 837–47.

Fu, C. Y., W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. 2017. "DSSD : Deconvolutional Single Shot Detector." arXiv preprint arXiv:1701.06659.

Gao, J., C. Wen, and M. Liu. 2017. "Robust Small Target Co-Detection from Airborne Infrared Image Sequences." *Sensors (Switzerland)* 17(10), pp. 1–21.

Girshick, R.. 2015. "Fast R-CNN." *Proceedings of the IEEE International Conference on Computer Vision* 2015 Inter, pp. 1440–48.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 580–87.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2016. "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(1):142–58.

Graves, A., M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. 2008. "A Novel Connectionist System for Unconstrained Handwriting Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(5), pp. 855–68.

Haala, N. and M. Kada. 2010. "An Update on Automatic 3D Building Reconstruction." *ISPRS Journal of Photogrammetry and Remote Sensing* 65(6), pp. 570–80.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem, pp. 770–78.

Hou, Q., M. M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr. 2019. "Deeply Supervised Salient Object Detection with Short Connections." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(4), pp. 815–28.

Hu, H., L. Wang, M. Zhang, Y. Ding, and Q. Zhu. 2020. "Fast and Regularized Reconstruction of Building Fac Ades from Street-View Images Using Binary Integer Programming." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 5(2), pp. 365–71.

Jiao, L., F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu. 2019. "A Survey of Deep Learning-Based Object Detection." *IEEE Access* 7(3), pp. 128837–68.

Khan, A., A. Sohail, U. Zahoora, and A. S. Qureshi. 2020. "A Survey of the Recent Architectures of Deep Convolutional Neural Networks." *Artificial Intelligence Review* 53(8), pp. 5455–5516.

Kong, T., A. Yao, Y. Chen, and F. Sun. 2016. "HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem, pp. 845–53.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." Pp. 1097–1105 in *Advances in neural information processing systems*.

Kakillioglu, B., S. Velipasalar, and T. Rakha. 2018. "Autonomous Heat Leakage Detection from Unmanned Aerial Vehicle-Mounted Thermal Cameras." *In Proceedings of the 12th International Conference on Distributed Smart* Cameras, pp. 1-6.

Li, G. and Y. Yu. 2016. "Visual Saliency Detection Based on Multi-scale Deep CNN Features." *IEEE Transactions on Image Processing* 25(11), pp. 5012–24.

Lin, T. Y., P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie. 2017. "Feature Pyramid Networks for Object Detection." *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125.

Liu, T. and A. Abd-Elrahman. 2018. "Deep Convolutional Neural Network Training Enrichment Using Multi-View Object-Based Analysis of Unmanned Aerial Systems Imagery for Wetlands Classification." *ISPRS Journal of Photogrammetry and Remote Sensing* 139, pp. 154–70.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. 2016. "SSD: Single Shot Multibox Detector." *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9905 LNCS, pp. 21–37.

R-CNN, Mask, Ross Girshick, Kaiming He, Georgia Gkioxari, and Piotr Doll. 2020. "Mask R-CNN." 42(2), pp. 386–97.

Rakha, T., A. Liberty, A. Gorodetsky, B. Kakillioglu, and S. Velipasalar. 2018. "Heat Mapping Drones: An

Autonomous Computer-Vision-Based Procedure for Building Envelope Inspection Using Unmanned Aerial Systems (UAS)." *Technology|Architecture + Design* 2(1), pp. 30–44.

Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem, pp. 779–88.

Redmon, J., and A. Farhadi. 2018. "YOLOv3: An Incremental Improvement." *arXiv preprint arXiv:1804.02767*.

Ren, Y., C. Zhu, and S. Xiao. 2018. "Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN." *Applied Sciences (Switzerland)* 8(5).

Sermanet, P., D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. 2014. "Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks." *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

Shen, Y., R. Ji, C. Wang, X. Li, and X. Li. 2018. "Weakly Supervised Object Detection via Object-Specific Pixel Gradient." *IEEE Transactions on Neural Networks and Learning Systems* 29(12), pp. 5960–70.

Shlezinger, N., N. Farsad, Y. C. Eldar, and A. J. Goldsmith. 2020. "ViterbiNet: A Deep Learning-Based Viterbi Algorithm for Symbol Detection." *IEEE Transactions on Wireless Communications* 19(5), pp. 3319–31.

Sun, G., P. Lin, and Y. Li. 2021. "Study on Improved YOLO_v3-Based Algorithm for Identifying Open Windows on Building Facades." *Journal of Physics: Conference Series* 1769(1).

U.S. Department of Energy. 2014. *Windows & Building Envelopes R&D: Roadmap for Emerging Technologies*. https://dl.acm.org/doi/10.5555/3390098.3390121

Viola, P. and M. Jones. 2001. "Rapid Object Detection Using a Boosted Cascade of Simple Features." Pp. 1193–97 in *Computer Vision and Pattern Recognition*.

Zhu, X., S. Lyu, X. Wang, and Q. Zhao. 2021. "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-Captured Scenarios." *Proceedings of the IEEE International Conference on Computer Vision* 2021-October, pp. 2778–88.

Zhu, J. Y., T. Park, P. Isola, and A. A. Efros. 2017. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks." *Proceedings of the IEEE International Conference on Computer Vision* 2017-Octob, pp. 2242–51.

AUTHOR BIOGRAPHIES

NORHAN BAYOMI is post-Doctoral Fellow at the MIT Environmental Solutions Initiative and a Research Assistant in the Urban Metabolism Group. She received her MSc in Building Technology from MIT in 2017, and PhD from the same program in 2021. Norhan's research focuses on the applications of AI in mapping climate change risks in cities and improving building energy simulation workflows.
nourhan@mit.edu

MOHANNED EL KHOLY is a computer scientist and fresh graduate of Computation and Cognition program at MIT. Mohanned's research focuses on the applications of computer vision techniques in building energy performance and climate change risk mapping.
mohanned@mit.edu

JOHN E. FERNANDEZ is the director of MIT Environmental Solutions Intiative and a professor of building technology in the Department of Architecture and a practicing architect. Fernández founded and directs the MIT Urban Metabolism Group, a highly multidisciplinary research group focused on the resource intensity of cities and design and technology pathways for future urbanization.

fernande@mit.edu

SENEM VELIPASALAR is a professor at the Department of Electrical Engineering and Computer Science in Syracuse University. Senem received the Ph.D. and M.A. degrees in Electrical Engineering from Princeton University in 2007 and 2004, respectively, the M.S. degree in Electrical Sciences and Computer Engineering from Brown University in 2001, and the B.S. degree in Electrical and Electronic Engineering with high honors from Bogazici University in 1999.
svelipas@syr.edu

TAREK RAKHA is an Assistant Professor of Architecture in Georgia Tech and is the Director of the High Performance Building Lab (HPBL). His research and teaching aims to transform climate-responsive and sustainable architectural design decisions to advance racially and socially just solutions using robotics, sensing, modeling, machine learning, and Artificial Intelligence (AI).
rakha@design.gatech.edu