

**FINAL YEAR PROJECT 2021-2022**  
**TEAM 18**

**Building extraction and roof type  
classification of aerial images for maximal  
PV panel installation**

**PROJECT GUIDE**  
Prof Dr. P. Uma Maheswari

**TEAM MEMBERS**  
Shruthi M - 2018103592  
Gayathri M - 2018103535  
Jayapriya M - 2018103029

## OVERALL OBJECTIVES

To simulate the **placement of PV panels** on rooftop of buildings from aerial images based on roof type for maximum energy consumption, the following steps are to be performed:

- ★ To **segment and detect buildings** in a given satellite image using **MultiRes UNet model** and perform background subtraction to extract the rooftops.
- ★ To **label** the extracted building rooftops into different classes - flat, gable, and hip and create a dataset.
- ★ To train convolutional neural nets and transfer learning models for **roof type classification** and perform a comparative study.
- ★ To perform **edge detection** on the extracted rooftops and mark boundaries to provide a modular layout of PV panel simulation.

# INTRODUCTION

- ★ Climate change has become a global concern and harnessing renewable resources is the way to construct a sustainable environment. Potential tapping of solar power for generating electricity has gained enormous popularity and people are increasingly gravitating toward the PV revolution.
- ★ However, traditional approaches, such as online assessment of rooftops are time-consuming and expensive. By automating the process of building roof extraction for PV panel placement, a lot of money and time can be saved.
- ★ A 3-step mechanism is proposed as a solution to address this. We use the AIRS dataset that provides a wide coverage of aerial imagery of Christchurch with 7.5 cm resolution for this purpose.
- ★ The first stage is building detection from aerial satellite images. We propose a deep learning framework called MultiRes UNet, with ResPath skip connections between the encoder and decoder structure. Better segmentation of buildings leads to maximum solar potential of rooftop areas.
- ★ This is followed by rooftop classification from the extracted buildings with different SOTA models as roof classification is used in new building design, retrofitting existing roofs, and efficient solar integration on building rooftops.
- ★ Finally, a simulation of modular layout of PV panels is produced considering the type of roofs, the tilt angle, length and width as provided by the user.

# LITERATURE SURVEY

S.NO	CITATION	METHODOLOGY	ADVANTAGES	LIMITATION
1.	An aerial image segmentation approach based on enhanced multi-scale convolutional neural network, 2019  Xiang Li, Yuchen Jiang, Hu Peng and Shen Yin in 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)	1. Segmentation model is performed using an encoder-decoder architecture.  2. A U-Net is constructed as the main network, and the bottom convolution layer of U-Net is replaced by a set of cascaded dilated convolution with different dilation rates.  3. Add an auxiliary loss function after the cascaded dilated convolution.	1. From the aspect of design and training, the approach does not involve manual features and does not require specific preprocessing or post-processing, which can reduce the influence of subjective factors.  2. The auxiliary loss function helps to make the network converge faster and optimize.	1. Segmentation of large buildings work well but boundaries are misaligned.  2. The bulges on the boundaries are lost and edges are not detected properly.  3. The algorithm performs well in one of the subset (countryside and forest) but does not perform well when tested on a different subset.

# LITERATURE SURVEY

S.NO	CITATION	METHODOLOGY	ADVANTAGES	LIMITATION
2.	<p>Deep learning based roof type classification using very high resolution aerial imagery, 2021</p> <p>M. Buyukdemircioglu , R. Can , S. Kocaman in The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLIII-B3-2021 XXIV ISPRS Congress</p>	<ol style="list-style-type: none"><li>1. Using UltraCam Falcon large-format digital camera orthophotos with 10cm spatial resolution is captured and roofs are manually classified into 6 different labels.</li><li>2. Data augmentation is applied and a shallow CNN architecture is trained.</li><li>3. The prediction is investigated by comparing with three different pre-trained CNN models, i.e. VGG-16, EfficientNetB4, and ResNet-50.</li></ol>	<ol style="list-style-type: none"><li>1. Simple CNN model are hence easier to implement.</li><li>2. Requires nominal hardware specifications.</li><li>3. The shallow CNN model has achieved 80% accuracy.</li></ol>	<ol style="list-style-type: none"><li>1. Since the roof images were clipped automatically from the orthophotos, there are few buildings with overlap.</li><li>2. Half-hip roofs are not classified properly and F1 score obtained for them is very low.</li><li>3. Different hyperparameter tuning was not done for the shallow CNN architecture.</li></ol>

# LITERATURE SURVEY

S.NO	CITATION	METHODOLOGY	ADVANTAGES	LIMITATION
3.	<p>Deep Convolutional Neural Network Application on Rooftop Detection for Aerial Imagery, 2019</p> <p>Mengge Chen, Jonathan Li, in Journal of Computational Vision and Imaging Systems</p>	<ol style="list-style-type: none"> <li>It is primarily based on Mask R-CNN with 3 stages.</li> <li>Feature extraction is based on existing deep learning model.</li> <li>RPN (Regional Proposal Network) is used to find RoI.</li> <li>Object classification is then performed.</li> </ol>	<ol style="list-style-type: none"> <li>Efficient and feasible approach to extract detached house from aerial images.</li> <li>RoIAlign method is used instead of RoIPool for better feature extraction.</li> </ol>	<ol style="list-style-type: none"> <li>Training data was less and hence less accuracy.</li> <li>Edges of the building are not detected properly.</li> <li>Comparatively less precision with other new state of art models.</li> </ol>
4.	<p>Solar Potential Analysis Of Rooftops Using Satellite Imagery, 2019</p> <p>Akash Kumar, Delhi Technology University, in ArXiv abs/1812.11606</p>	<ol style="list-style-type: none"> <li>Dataset is manually collected for India.</li> <li>Adaptive Edge Detection and Contours are focused to segment out rooftop boundaries and obstacles present inside them along with polygon shape approximation.</li> </ol>	<ol style="list-style-type: none"> <li>Provides a comparative analysis of the solar potential of the building.</li> <li>Several types of the rooftop are considered to learn the intra-class variations.</li> </ol>	<ol style="list-style-type: none"> <li>The resolution of satellite image is very poor hence segmentation is not done properly.</li> <li>There are some outliers that are plotting solar panels outside the building rooftop area.</li> </ol>

# LITERATURE SURVEY

S.NO	CITATION	METHODOLOGY	ADVANTAGES	LIMITATION
5.	<p>Understanding rooftop PV panel semantic segmentation of satellite and aerial images for better using machine learning, 2021</p> <p>Peiran Li , Haoran Zhang , Zhiling Guo, in Advances in Applied Energy, Volume 4, 100057, ISSN 2666-7924</p>	<p>1. Data pre-processing involves collecting patch satellite images from Google for the city of Heilbron and manually labelling them.</p> <p>2. Object proportion distribution in image-level and object occurrence possibility at pixel level is statistically analysed.</p> <p>3. SOTA PV segmentation model (DeepSolar) is used to extract visual features.</p> <p>4. Local Binary Pattern (LBP) is used for texture feature extraction &amp; color histograms for color feature extraction.</p>	<p>1. Class imbalance of PV and non-PV panels in rooftops is resolved by hard sampling, soft sampling.</p> <p>2. The homogenous textural feature using LBP has served as an additional part for some easily confused cases to improve the robustness of the model.</p>	<p>1. IOU is less than the acceptable range (0.5) for 1.2m resolution images.</p> <p>2. Patch overlapping occurs while stitching the tiles and this leads to segmentation errors.</p> <p>3. Lighting conditions resulted in different clustering groups in color clustering of PV/Non-PV and has led to misclassification.</p>

## SUMMARY OF ISSUES IN LITERATURE SURVEY

The frameworks and models discussed above for image segmentation have made progress in addressing the issue of building segmentation to a great extent but there are certain major flaws in most of the cases.

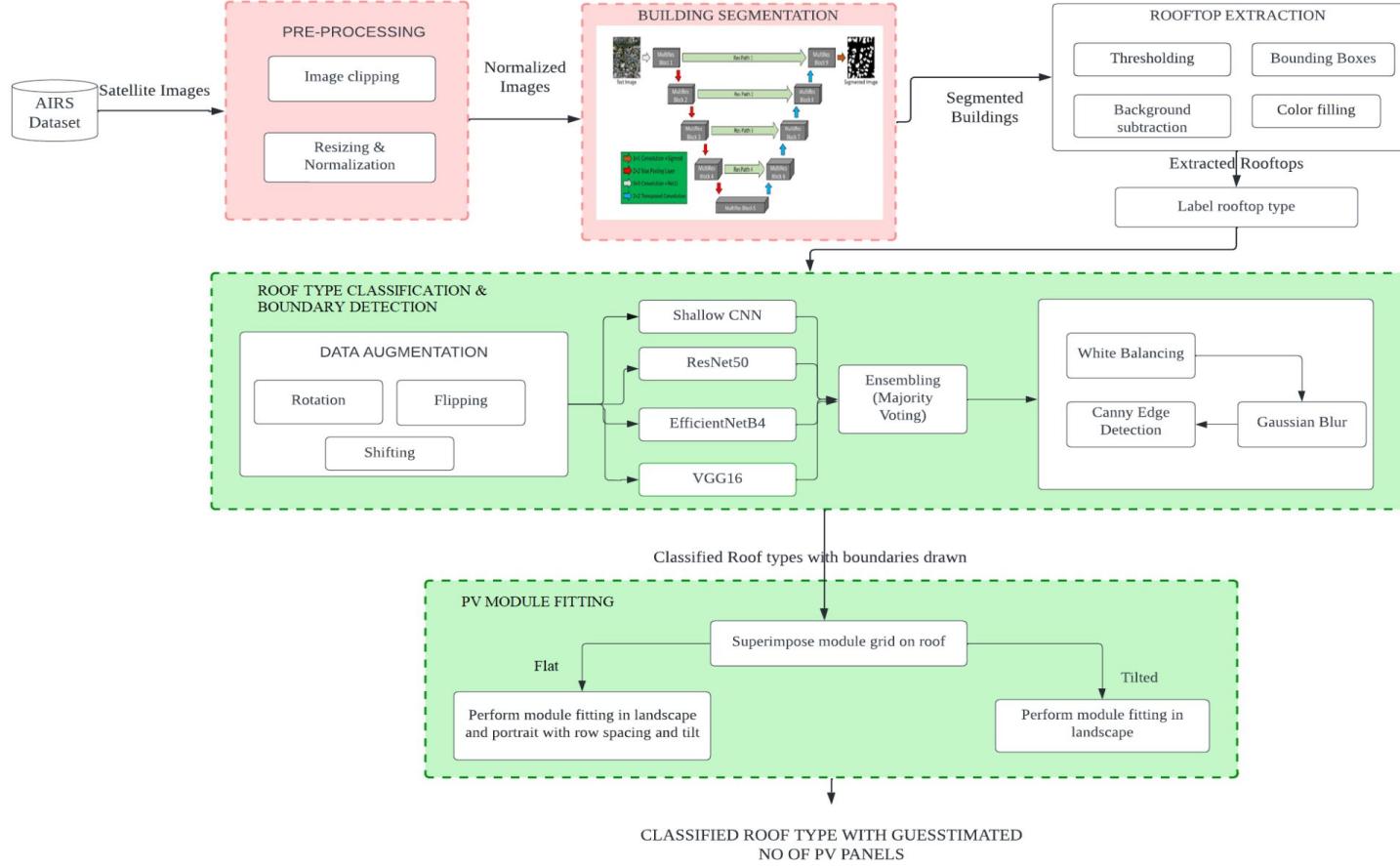
- ★ In situations where shadows, vegetation and parking lots and other obstacles enclose buildings, the above frameworks showed poor results in building detection.
- ★ In addition, there was no clear distinction to demarcate adjacent buildings and boundaries of buildings were not properly detected.
- ★ Furthermore, when a satellite or aerial image spanning a large area with several buildings was provided, only manual cropping of roof type images was done for classification, and roof tops were not retrieved automatically in any of the approaches outlined above.
- ★ Some of the studies mentioned above calculate the discrete number of PV modules that should be placed on the available roof space, but none of them computed the actual modular layout that would fit on each roof segment.
- ★ Shadows and other obstacles in the above studies are often neglected while considering the placement of PV panels which results in bad prediction of energy consumption.

## PROPOSED SYSTEM

Our proposed system aims to do the following:

- ★ **Automate** the time-consuming and labor-intensive process of manual assessment of roof tops for classification and for PV panel installation by going through a **3-stage pipeline**.
- ★ Use **SOTA MultiRes UNet** to perform building segmentation and extract rooftops using background subtraction given a large satellite image.
- ★ Train different classifier models and perform ensembling (majority voting) to **classify** the different type of roofs.
- ★ Provide a **simulation** of placement of PV panels on roof top.

# OVERALL ARCHITECTURE



# OVERALL ARCHITECTURE

- ★ The pipeline consists of 3 modules: (i) **Building segmentation and detection**, (ii) **Roof type classification and boundary detection**, and (iii) **PV module fitting**.
- ★ The first module mainly emphasizes building detection and rooftop extraction from satellite images.
  - The first stage of building segmentation includes two pre-processing steps: clipping satellite images and the accompanying ground truth masks, followed by scaling and normalization.
  - The normalized images are then trained using the MultiRes UNet architecture for segmenting the buildings.
  - Image processing techniques like finding contours, drawing bounding boxes and background subtraction are performed to extract the rooftops of buildings.
- ★ Second module focuses on roof type classification and boundary detection.
  - We manually label the extracted rooftops into 3 classes: Flat, Gable, Hip which are fed to four different models and a comparative analysis is made.
  - Majority voting is used to combine predictions and improve classification accuracy.
  - Following that, edge detection of rooftops takes place to mark boundaries on rooftops.
- ★ The final module resorts to providing a simulation of modular layout of PV panels fitted on top of roofs.

## LIST OF MODULES

MODULE I: BUILDING DETECTION

MODULE II: ROOF TYPE CLASSIFICATION AND BOUNDARY  
DETECTION

MODULE III: MAXIMAL FITTING ALGORITHM

# MODULE DESIGN

## MODULE I: BUILDING DETECTION

**INPUT:** AIRS Dataset

**OUTPUT:** Rooftops of different buildings

- ★ The first step involves pre-processing with clipping of large aerial images into smaller tiles and performing resizing and normalization.

Input: Aerial images .

Output: Smaller patches of normalized images.

- ★ Following this, the MultiRes UNet architecture is implemented.

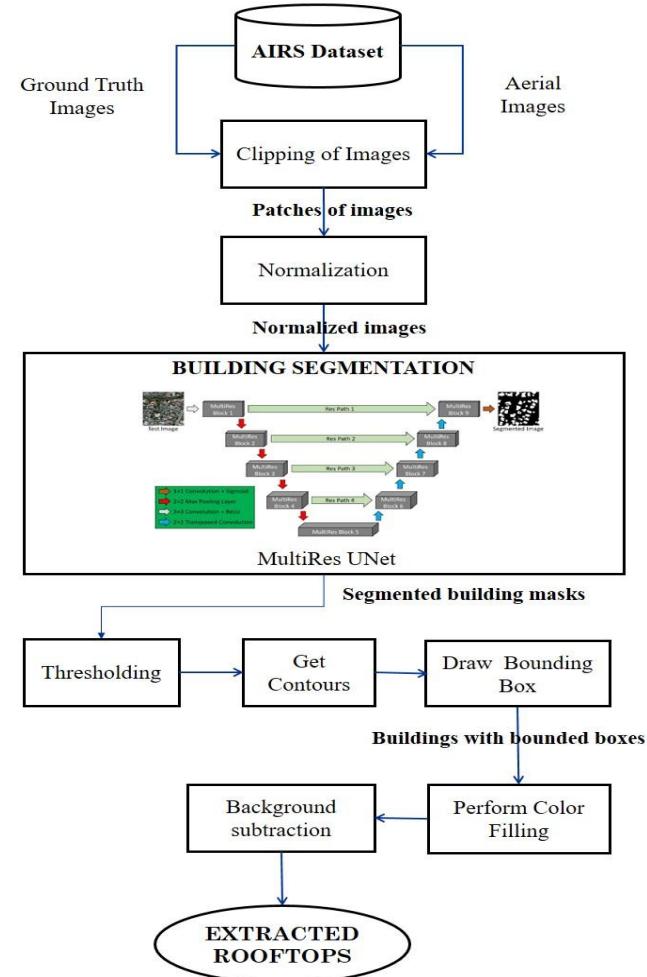
Input: Ground truth mask patches and aerial image patches for training.

Output: Trained model with binary mask for buildings

- The final step involves background subtraction.

Input: Binary mask of buildings.

Output: Rooftops of different buildings.



# PSEUDO-CODE

Begin

## Pre-Processing

- a. Get the original dimensions and the dimensions, stride of smaller patches.
- b. Find overlapping areas.  $\text{overlapping} := (\text{size} / \text{stride}) - 1$
- c. for  $i$  in range ( $\text{orig\_shape} / \text{stride} - \text{overlapping}$ ):
- d. Crop from  $i*\text{stride}:i*\text{stride}+\text{size}$ ,  $j*\text{stride}:j*\text{stride}+\text{size}$
- e. Perform resizing and normalization.

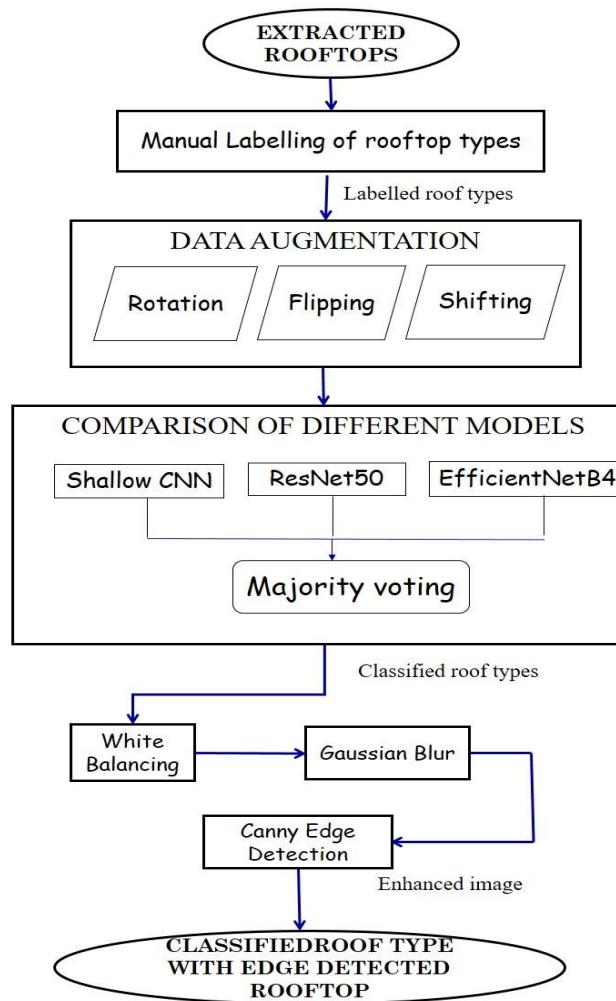
## Training the MultiRes UNet Architecture

- f. Split dataset into train - 1548 images, validation - 72 & testing - 144
- g. Define the MultiRes model.
- h. do:
  - i. Tweak hyperparameters: (lr, batch size, epochs, optimizer, loss).
  - ii. Train the model.
  - iii. Plot graphs and check on validation data.
- i. while (find the best model)
- j. Save weights for the best model.

## Background Subtraction

- k. Convert RGB to grayscale.
- l. Find contours.
- m. Find the bounding box coordinates  $:= (x, y), (x+w, y), (x, y+h), (x+w, y+h)$
- n. Perform color filling.
- o. Remove background from foreground buildings.

End



## MODULE DESIGN

### MODULE II: ROOF TYPE CLASSIFICATION & BOUNDARY DETECTION

**INPUT:** Extracted rooftops.

**OUTPUT:** Classified roof type with edge detected rooftop.

- ★ The first step is manually labeling the rooftops into 3 different classes - Flat, Gable, Hip.
- ★ This is followed by data augmentation with rotation, flipping and shifting as transformations.
- ★ Following this, we try 4 different models - customized CNN, ResNet50, EfficientNetB4 and VGG16 perform majority voting  
 Input: Rooftops with labels.  
 Output: Classified rooftop type.
- ★ The final step involves boundary detection performed by White Balancing, Gaussian blur and Canny edge detection.  
 Input: Classified rooftop type.  
 Output: Boundary detected rooftops.

# PSEUDO-CODE

Begin

## Data Augmentation

- a. Rotation - Randomly generate an angle (theta) and rotate the image by (theta) degrees clockwise and anticlockwise.
- b. Shifting - ImageDataGenerator (width\_shift\_range = 0.10)
- c. Flipping - ImageDataGenerator (horizontal\_flip = True)

## Training different deep learning models

- d. Split the dataset into train - 80%, validation - 10% and testing- 10%.
- e. Define a customized CNN model.
- f. Train:
  - i. Tweak hyperparameters (Learning rate, batch size, and optimizer).
  - h. Use pre-trained ResNet50, EfficientNetB4 and VGG16.
  - i. Fine tune the pre-trained models.
  - j. Repeat above steps for the other 2 transfer learning models.

j. Repeat above steps for the other 2 transfer learning models.

k. For an unseen image:

- i. Tweak hyperparameters (Learning rate, batch size, and optimizer).
- ii. Perform majority voting to identify the class that gets the maximum votes.

## Boundary Detection

- l. White patching - Set a percentile score and remove haze from the image.
- m. Gaussian Blur - Try for different kernel size and  $\sigma$ .
- n. Find the threshold for adaptive canny edge technique.

End

---

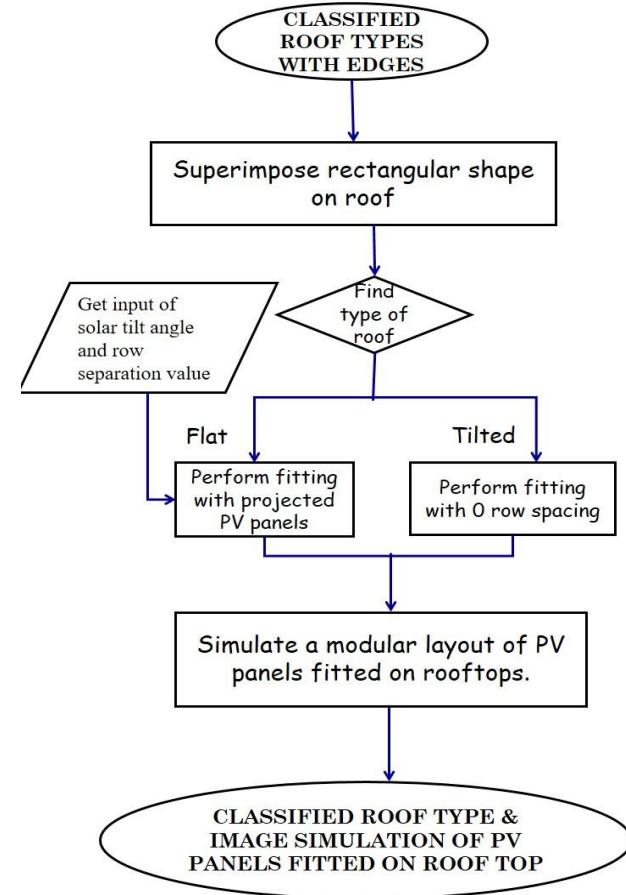
# MODULE DESIGN

## MODULE III: PV MODULE FITTING

**INPUT:** Classified roof type with edge detected rooftop .

**OUTPUT:** Classified roof type with simulation of PV panel on rooftops.

- ★ The boundary detected rooftops are superimposed with rectangular PV module shape.
- ★ Based on the type of roof:
  - For flat roofs: tilt angle in the range ( $20^\circ - 50^\circ$ ) and row spacing is received from user and module fitting is performed.
  - For tilted roofs: tilt angle in the range ( $0^\circ - 10^\circ$ ) with zero row spacing is used to perform module fitting.
- ★ An image of simulation with PV panels on top of roof is the final output.



## PSEUDO-CODE

Begin

1. Get dimensions of PV panel from user (length, width, and spacing).
  2. Get tilt angle based on the type of roof.
    - a. For flat roofs, tilt angle := (20° - 50°)
    - b. For tilted roofs, tilt angle := (0° - 10°)
  3. Create a rectangular module grid according to the settings specified for flat and tilted roofs.
  4. do:
    - I    a. Identify new points after projecting the tilted PV panels.  
 $(x', y') := (x\cos - y\sin, x\sin + y\cos)$
    - b. Identify the contour area.
    - c. if area < 5:  
Move to the next point as this area could be an obstacle or smaller patch unsuitable for panel placement.
    - else:  
Fix a PV panel on the identified points.
  - d. if flat:  
Shift the module grid with specified row spacing.
  - else:  
Shift the module grid with 0 spacing.
  - e. Move to the next point on the line segment and fix panel over there.
- while (entire roof top is analyzed)
- End

# IMPLEMENTATION

## DATASET

AIRS dataset covering the full area of Christchurch in New Zealand with satellite images and ground truth masks.

Training set: 857 images

Validation Set: 90 images

Testing Set: 90 images

Spatial dimensions:  $10000 \times 10000$  pixels

Spatial resolution: 7.5 cm

## EXPERIMENTAL SETUP

- ★ Tensorflow backend.
- ★ Keras Framework.
- ★ Environment used: Colab Pro with 16BG, T4 GPU and Kaggle GPU notebooks
- ★ Frontend integrated with flask.

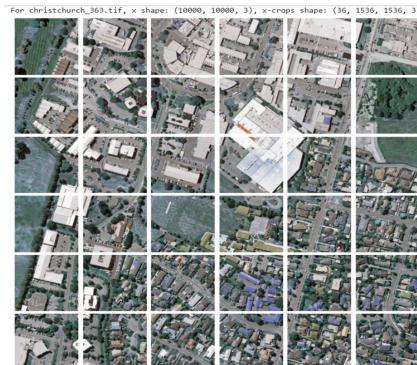
# MODULE I - BUILDING DETECTION

## 1.1 Clipping original aerial images

- ★ Direct segmentation of large satellite images is difficult.
- ★ Dimensions of  $10000 * 10000$  pixels is cut into 36 smaller patches of size  $1536 * 1536$  by sliding window technique.



Aerial satellite image



Patches of image after clipping

## 1.2 Clipping corresponding mask images

- ★ Corresponding ground truth masks for satellite images are also clipped.



Ground truth segmented image



Ground truth patches after clipping

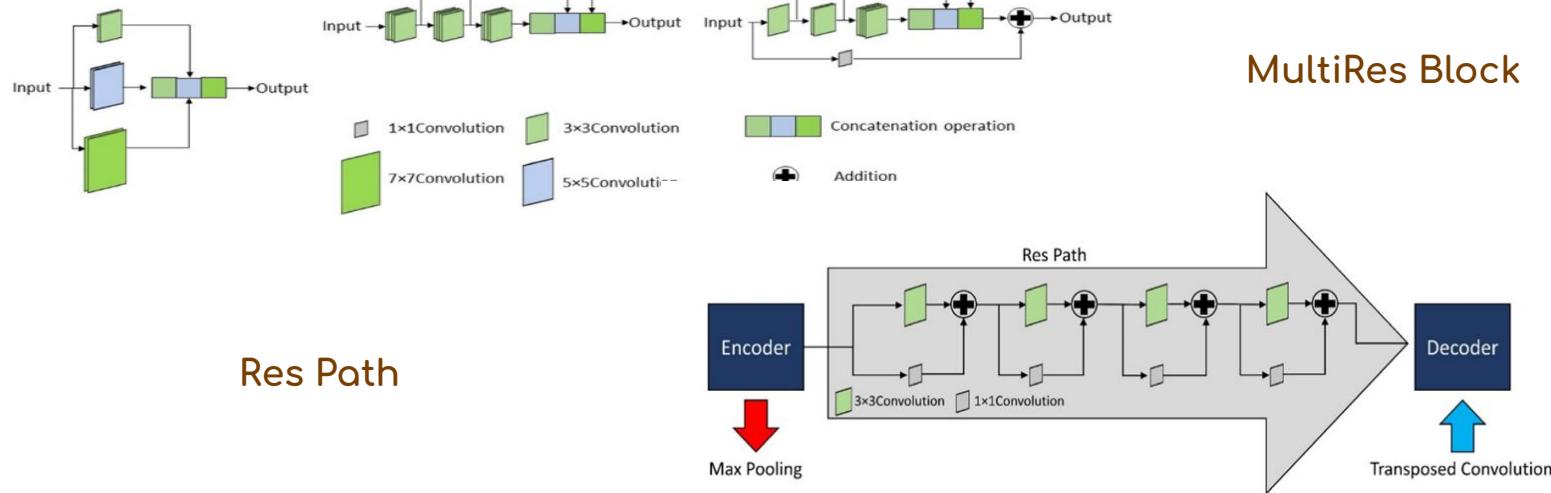
# MODULE I - BUILDING DETECTION

## 1.3 Resizing and Normalization

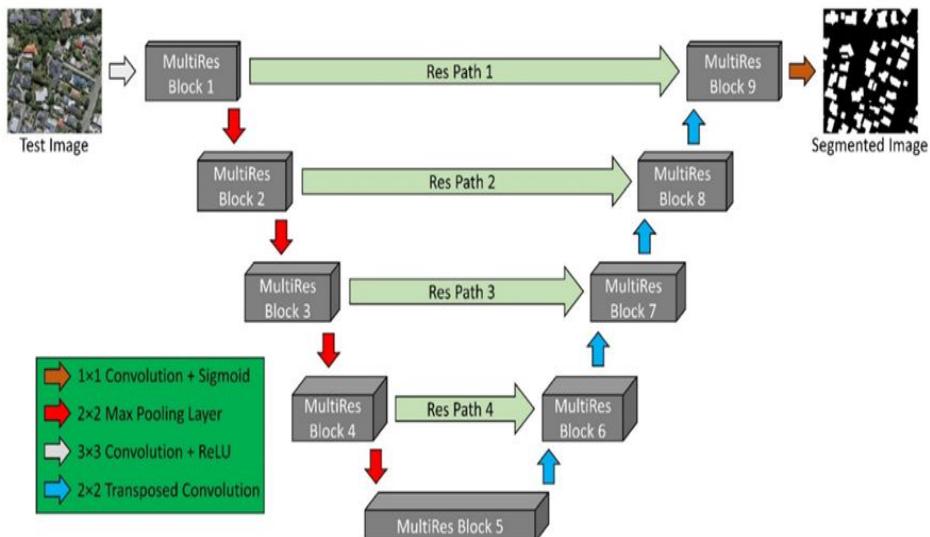
Images of size 1536 \* 1536 are difficult to train owing to hardware constraints. Images are hence resized to 256 \* 256 with INTER\_CUBIC interpolation method for better resolution of images.

## 1.4 Adapting the MultiRes UNet architecture

The architecture of the MultiRes UNet network consists of 2 important blocks: the MultiRes Block and the Res Path.



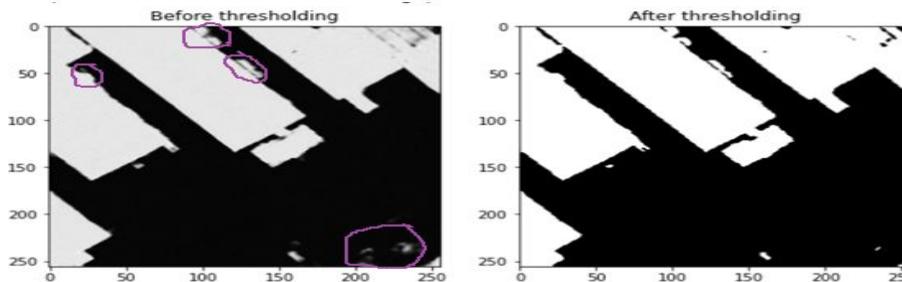
# Architecture of MultiRes UNet



Hyper parameters	Values
Learning rate	0.0001
Epochs	100
Batch size	8
Image dimensions	256 x 256
Optimizer	Adam
Loss	Binary cross entropy
Activation Function	ReLU (in all convolution layers) Sigmoid (in the output layer)

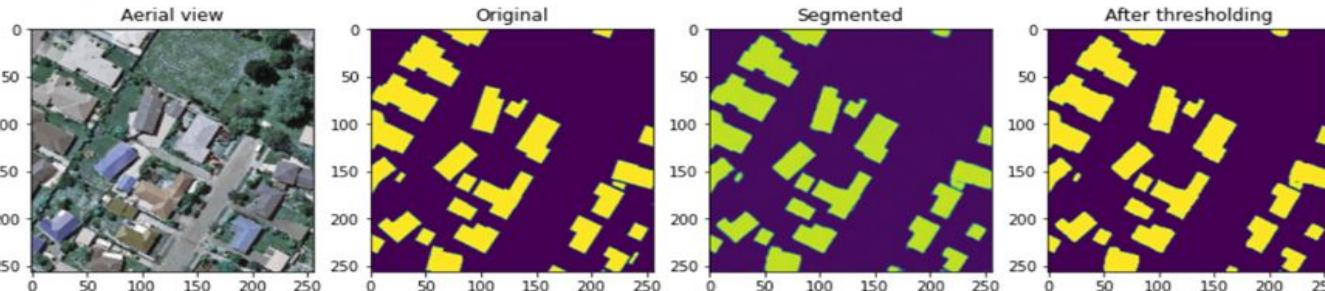
## 1.5 Applying threshold

Threshold is set to 0.5 to delineate the boundaries of buildings and differentiate the edges properly. The violet color markings show the difference before and after the threshold is applied.



Results of building segmentation before and after applying threshold.

Image number: 1  
IOU Score: 0.9527599215507507  
Dice Coefficient: 0.975807785987854  
MCC: 0.9687466621398926



Results of building segmentation with MultiRes UNet model

# MODULE I - BUILDING DETECTION

## 1.6 Rooftop Extraction

### 1.6.1 Get contours

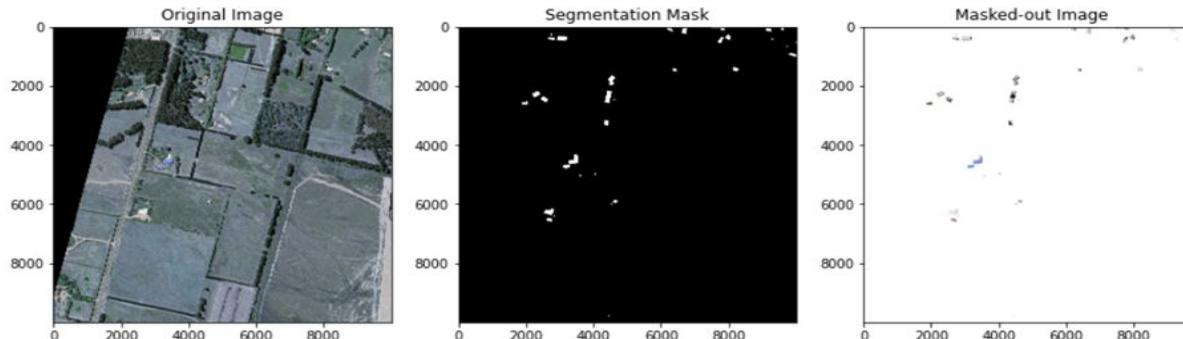
- ★ Contours are used to detect the borders of objects and localize them easily in an image.
- ★ Grayscale transformation and then binary thresholding is applied on it following which the contours of buildings are identified.

### 1.6.2 Drawing bounding box

- ★ Bounding boxes are used to highlight the regions of interest after obtaining contours from an image.

### 1.6.3 Background Subtraction

- ★ Overlay segmentation mask on top of aerial satellite image.
- ★ Subtract bounded box segmented regions detected in earlier step with satellite image.



Original satellite image (left), masked image after training on MultiRes UNet model (center), extracted rooftops (right)

## MODULE II - ROOF TYPE CLASSIFICATION AND BOUNDARY DETECTION

### 2.1 Preparation of dataset and labeling roof type

- ★ After conducting rooftop extraction in the preceding step, a dataset including 1115 rooftop photos is populated into three different categories: Flat, Gable, and Hip.
- ★ Manual labeling of dataset into 3 classes was performed.

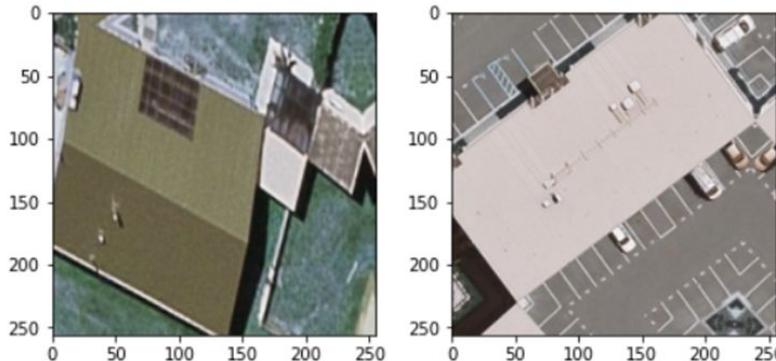
FLAT		
GABLE		
HIP		

Samples of images of each class from Christchurch, New Zealand

## MODULE II - ROOF TYPE CLASSIFICATION AND BOUNDARY DETECTION

### 2.2 Data Augmentation

- ★ Data augmentation is applied to increase the dataset size for better model training.
- ★ Rotation, shifting and flipping are the three main techniques used here.



Results of image after performing data augmentation

### 2.3 Roof type classification

- ★ Deep learning and transfer learning models are employed for roof type classification.
  - A. Shallow CNN
  - B. ResNet50
  - C. EfficientNetB4
  - D. VGG16
- ★ A comparative analysis of these models on the AIRS dataset is performed.
- ★ The model is validated with supervised roof type data from Potsdam.
- ★ To enhance the classification results, ensembling (majority voting) is used.

## 2.4 Boundary Detection

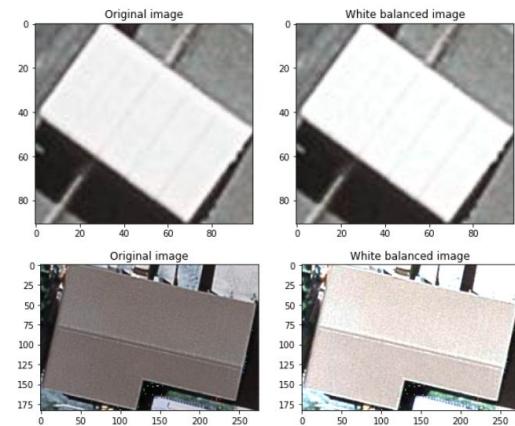
### 2.4.1 White Balancing

- ❑ Satellite images are usually under highly variable lighting conditions which cause haze.
- ❑ White balance is used to remove any coloured haze from an image so that it seems to be under white light.

### 2.4.2 Gaussian Blurring

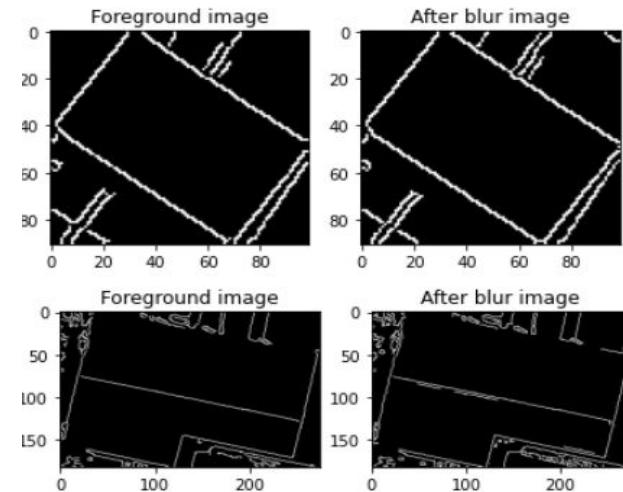
- ❑ Gaussian blurring reduces noise & accentuates edge contrast to identify boundaries on roof tops.
- ❑ A kernel size of 3X3 is used with  $\sigma = 1.5$

White balanced image



### 2.4.3 Auto Canny Edge Detection

- ❑ Multi-staged algorithm that includes noise reduction, finding intensity gradients and hysteresis suppression
- ❑ The median of single channel pixel intensity is calculated.
- ❑ The upper and lower threshold is calculated by using mean and variance of an image intensity.

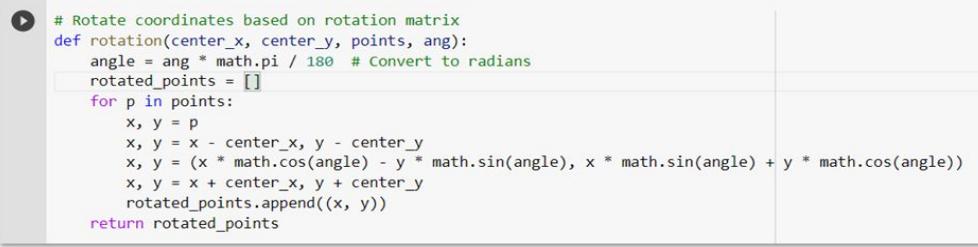


Boundaries after performing edge detection

# MODULE III - PV MODULE FITTING

## 3.1 Rotation of panels

- ★ PV panels that have zero tilt angle can be mounted directly on rooftops and moved.
- ★ For panels with non- zero tilt angle, projection of panels is necessary.  
Let us assume the initial coordinates of PV panels to be (x,y).  
Rotated coordinates  $(x', y') = (x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$ .



```
# Rotate coordinates based on rotation matrix
def rotation(center_x, center_y, points, ang):
    angle = ang * math.pi / 180 # Convert to radians
    rotated_points = []
    for p in points:
        x, y = p
        x, y = x - center_x, y - center_y
        x, y = (x * math.cos(angle) - y * math.sin(angle), x * math.sin(angle) + y * math.cos(angle))
        x, y = x + center_x, y + center_y
        rotated_points.append((x, y))
    return rotated_points
```

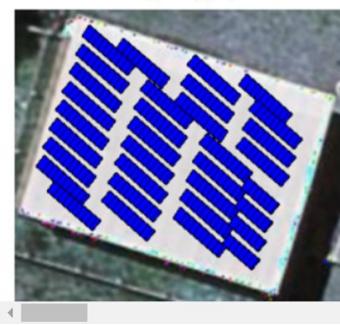
Implementation of  
rotation of points

## 3.2 PV Panel Placement

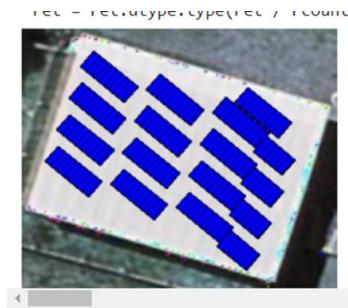
- ★ Contours are drawn for identifying area.
- ★ All contour areas with size less than 5 are eliminated assuming them to be obstacles.
- ★ Otherwise, the rotated points are checked to see if they lie within the boundary of rooftops.

### 3.2 PV Panel Placement

- ★ For each of the rotated panels, pointpolygontest is undertaken to ensure that all points of solar PV panels lie inside the rooftop.
- ★ If the rotated panel coordinates lie outside, then the panels are not placed.
- ★ Bresenham's algorithm is used to identify points in straight line segments so that PV panels can be placed one after the other along the line segment.
- ★ This process is continued by going through the entire roof top area.



PV panels with dimensions length - 20mm; tilt angle - 40; width - 10 mm (left) and 20mm (right)



Placement of PV panels when there are obstacles



# METRICS FOR EVALUATION

## For building detection segmentation:

### ❑ IoU - Intersection over Union / Jaccard Coefficient

To quantify the accuracy of our model for building detection, we use Jaccard coefficient which is to measure the similarity between detected regions and ground truth regions. Jaccard Similarity Index(JSI) measures the similarity for the two sets of pixel data, with a range from 0% to 100%. The higher the percentage, the more precise prediction. It is defined as follows:

$$JSI = \frac{r_d \cap r_g}{r_d \cup r_g}$$

where  $r_d$  denotes the masked region for building detection, and  $r_g$  indicates the groundtruth region for building segmentation.

### ❑ DICE Coefficient

We use DICE coefficient to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. DICE coefficient is 2 times the area of overlap divided by the total number of pixels in both the images. The formula is given by:

where X is the predicted set of pixels and Y is the ground truth.

$$\frac{2 * |X \cap Y|}{|X| + |Y|}$$

# METRICS FOR EVALUATION

## ❑ MCC - Matthews Correlation Coefficient

We use the MCC , a standard measure of a binary classifier's performance, where values are in the range -1.0 to 1.0, with 1.0 being perfect building segmentation, 0.0 being random building segmentation, and -1.0 indicating building segmentation is always wrong.

$$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

## ❑ Accuracy

Accuracy is the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

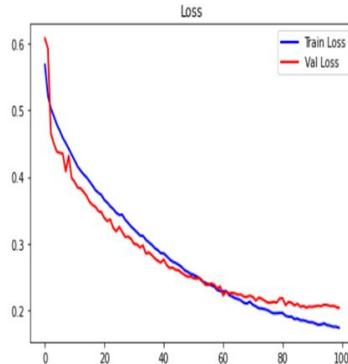
$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

## For roof type classification:

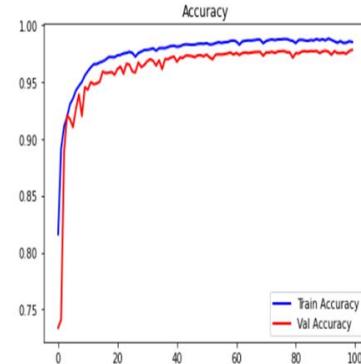
### ❑ Classification Report

The classification report is used to measure the quality of predictions from a classification algorithm. Precision, Recall and F1 scores are calculated on a per-class basis based on True Positives, True Negatives, False Positives, False Negatives. Here, we calculate the above values for each of the classes, namely: Flat, Gable, Complex.

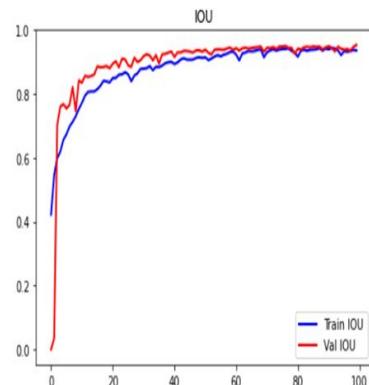
## PERFORMANCE RESULTS FOR BUILDING SEGMENTATION



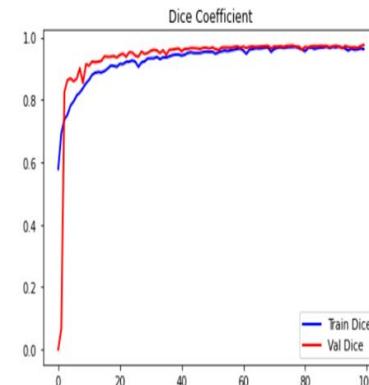
Model Loss



Model Accuracy



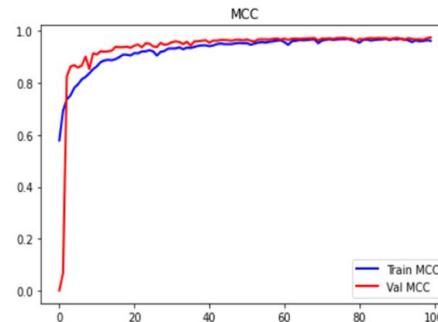
Model IOU



Model Dice Coefficient

Performance metrics	Training Set	Validation Set
IOU (%)	93.53	95.25
Dice Coefficient (%)	96.25	97.56
MCC (%)	95.87	96.74
Accuracy (%)	98.51	97.83
Loss	0.1734	0.2033

Performance results of the model on training and validation set

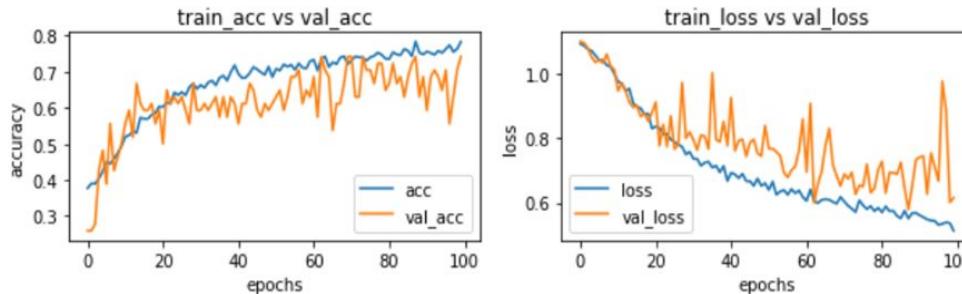


Model MCC

## PERFORMANCE INFERENCE FOR BUILDING SEGMENTATION

- ★ IoU, which penalizes heavily than dice coefficient, has achieved a value of 96.27% at the end of 100 epochs.
- ★ Average IoU after training 1548 images for 100 epochs comes to 93.53% and the validation IoU of 95.25% demonstrates that the model performs well even after the images are resized and normalized.
- ★ It can be seen that the Dice coefficient starts with 58% at the beginning of training and gradually attains a value of 98.5% with no major fluctuations.
- ★ Training and validation curve coincides with each other which indicates there is no overfitting.
- ★ The MCC value is close to 98% at the end of 100 epochs which indicates that buildings have been identified correctly.
- ★ The model has performed remarkably well as the training loss hovers around 0.173 and validation loss is close to 0.2. This is a reasonably good value which indicates that our model performs well in segmenting the buildings accurately.

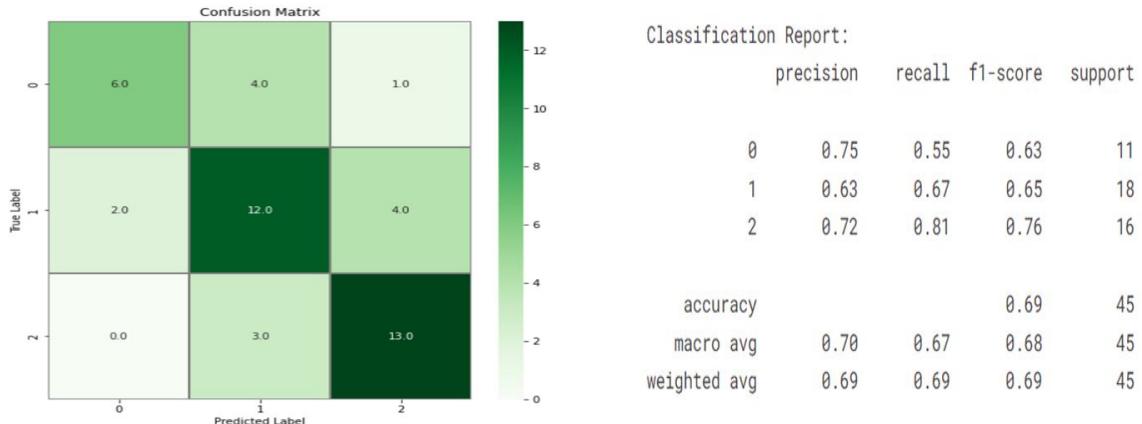
# CLASSIFICATION RESULTS FOR CNN MODEL



- ★ The CNN model achieved an accuracy of 78% on the training set and 74% on the validation set.

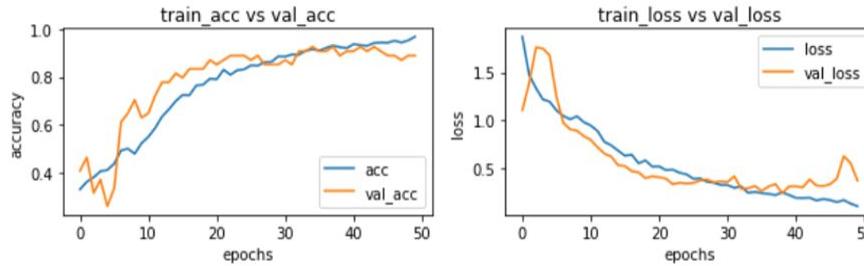
Graphs showing the training and validation accuracy and loss for CNN model

- ★ The confusion matrix reveals that gable and hip classes are accurately classified to a larger extent.
- ★ Hip class has the highest recall value of 81% followed by gable 62 (67%) and then flat (55%).



Confusion matrix and classification report on test data for CNN model

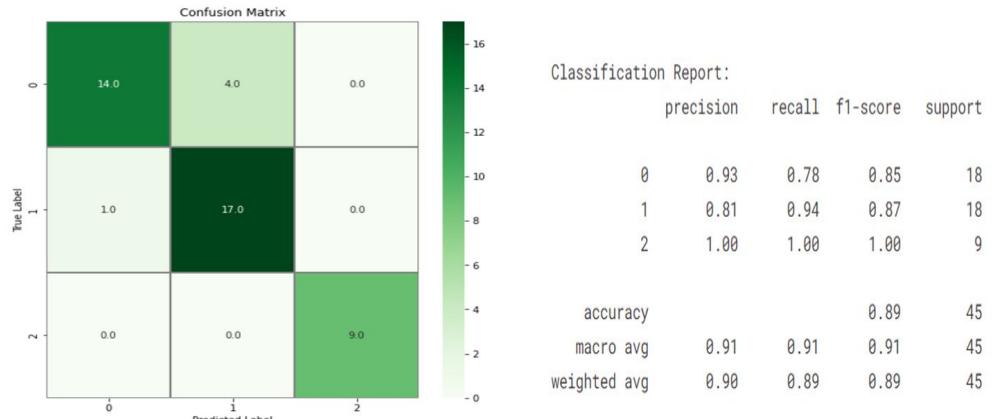
# CLASSIFICATION RESULTS FOR RESNET-50 MODEL



- ★ The ResNet-50 has yielded an accuracy of 91.59% on the training set and 92.5% on the validation set.
- ★ The loss is also very less at only 0.245 on the training set.

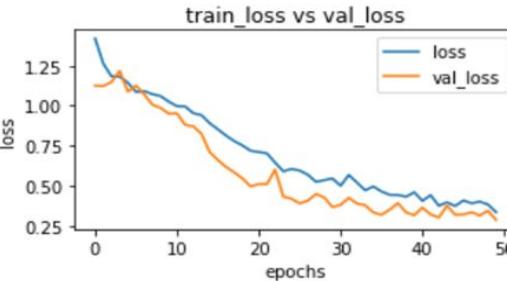
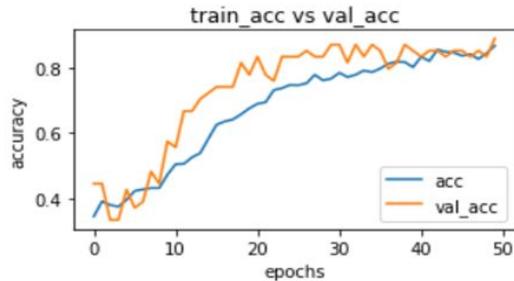
Graphs showing the training and validation accuracy and loss for ResNet50

- ★ Hip class is 100 % accurately classified.
- ★ The classification of flat classes is better compared to the previous CNN model.
- ★ The overall accuracy on the test samples is 89% with F-1 score averaging to 91%.



Confusion matrix and classification report on test data for ResNet-50 model

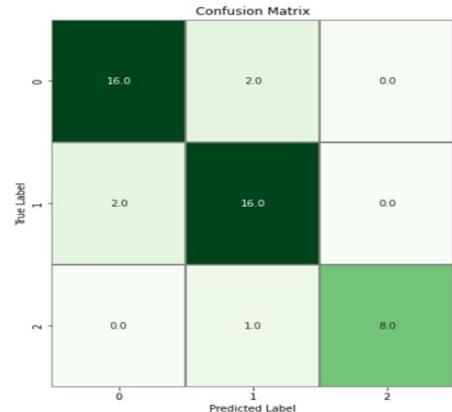
# CLASSIFICATION RESULTS FOR EFFICIENTNETB4 MODEL



- ★ The graphs exhibits that the accuracy of the model has steadily grown from 40% to 89% and the model loss has significantly reduced from 1.25 to 0.33 at the end of 50 epochs.

Graphs showing the training and validation accuracy and loss for EfficientNetB4

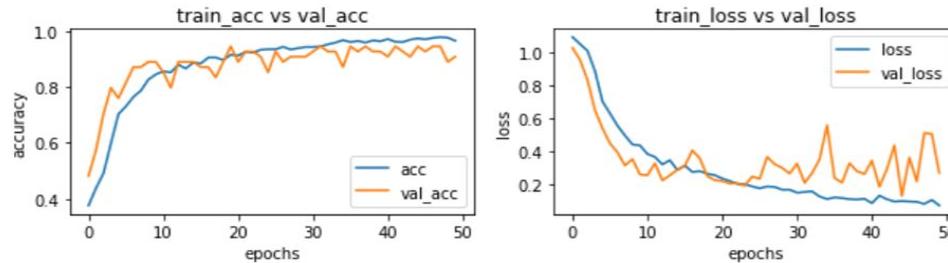
- ★ When compared with the ResNet50 model which had higher accuracy in correctly classifying hip class, EfficientNetB4 model has achieved higher F-1 score and accuracy in identifying flat roof tops.
- ★ The average F1 score, precision, and recall hover around 91%.



Classification Report:				
	precision	recall	f1-score	support
0	0.89	0.89	0.89	18
1	0.84	0.89	0.86	18
2	1.00	0.89	0.94	9
accuracy			0.89	45
macro avg	0.91	0.89	0.90	45
weighted avg	0.89	0.89	0.89	45

Confusion matrix and classification report on test data for EfficientNetB4

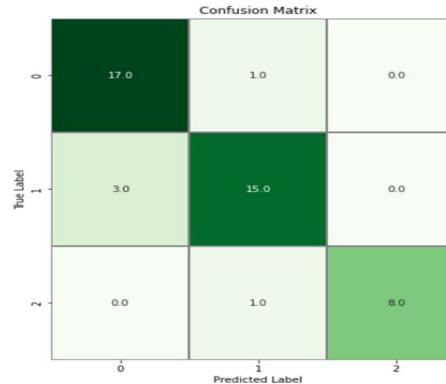
# CLASSIFICATION RESULTS FOR VGG16 MODEL



- ★ VGG-16 model has achieved the highest accuracy compared to the other three models.
- ★ The classification accuracy on the validation set is 94.45% while the training accuracy is around 97%.

Graphs showing the training and validation accuracy and loss for VGG-16

- ★ The F-1 score for flat class is the highest with VGG16 compared to other models.
- ★ The overall classification accuracy has totalled to 89%.



Classification Report:				
	precision	recall	f1-score	support
0	0.85	0.94	0.89	18
1	0.88	0.83	0.86	18
2	1.00	0.89	0.94	9
accuracy			0.89	45
macro avg	0.91	0.89	0.90	45
weighted avg	0.89	0.89	0.89	45

Confusion matrix and classification report on test data for VGG-16 model

## COMPARISON OF RESULTS WITH DIFFERENT MODELS

Model	Learning rate & Optimizer	Batch size	Loss	Accuracy (%)	F-1 score
Shallow CNN	0.0001 & RMSProp	16	0.7233 - Train 0.8067 - Test	79.25 - Train 69.8 - Test	0.69
ResNet- 50	0.00001 & RMSProp	8	0.2450 - Train 0.7123 - Test	92.59 - Train 88.89 - Test	0.91
EfficientNetB4	0.00001 & Adam	16	0.3359 - Train 0.343 - Test	86.70 - Train 89 - Test	0.91
VGG16	0.0001 & RMSProp	4	0.2021 - Train 0.4894 - Test	96.5 - Train 88.89 - Test	0.89

Results of different classification models on AIRS dataset

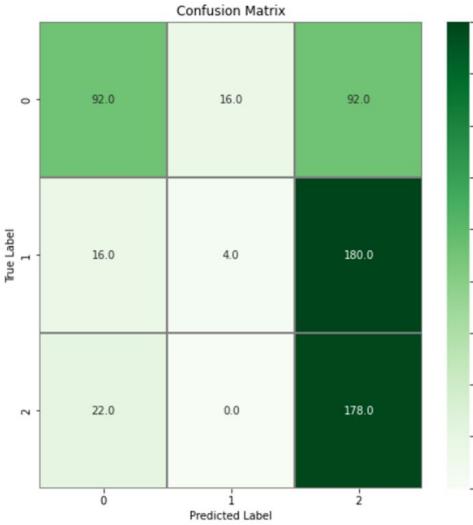
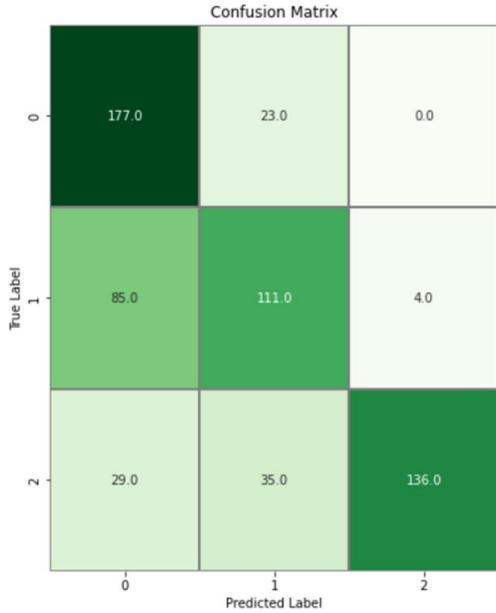
## VALIDATION OF MODELS ON POTSDAM DATASET

Model	Learning rate & Optimizer	Batch Size	Loss	Accuracy (%)	F-1 score
Shallow CNN	0.0001 & RMSProp	16	1.2829	45.67	0.38
ResNet- 50	0.00001 & RMSProp	8	0.9248	70.67	0.71
EfficientNetB4	0.00001 & Adam	16	0.7110	65.17	0.66
VGG16	0.0001 & RMSProp	4	0.8507	74	0.73

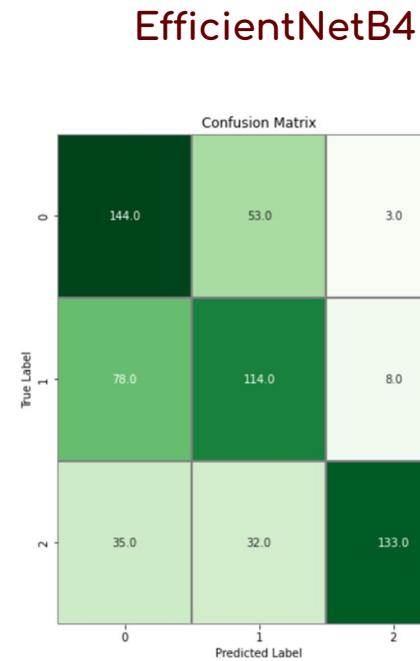
Results of testing our trained models on  
Potsdam dataset

# VALIDATION OF MODELS ON POTSDAM DATASET

ResNet50

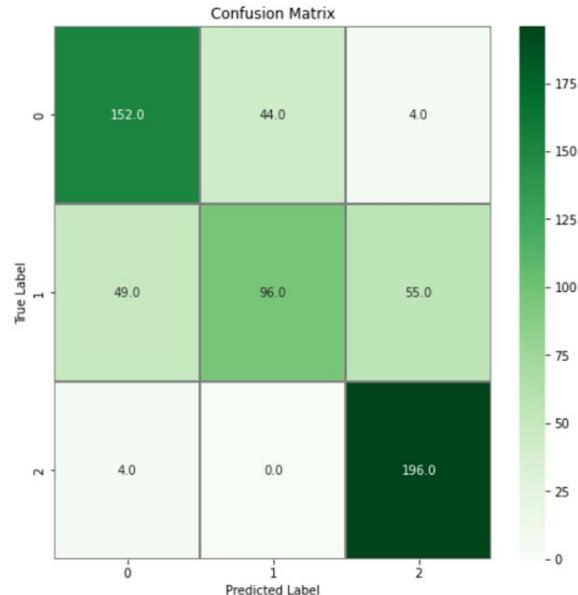


Shallow CNN

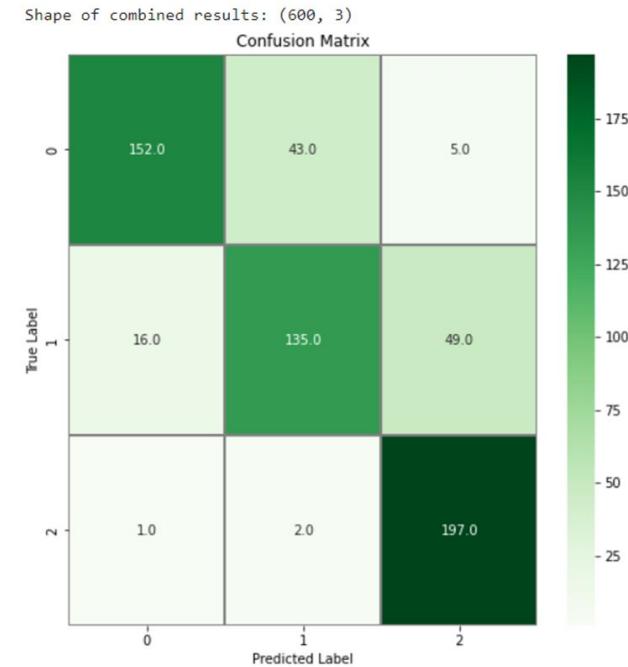


# VALIDATION OF MODELS ON POTSDAM DATASET

VGG16



Majority Voting



## COMPARATIVE ANALYSIS

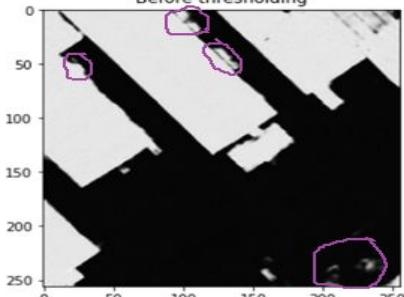
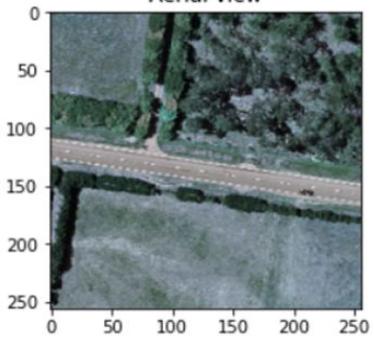
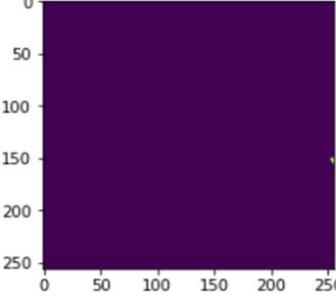
	MultiRes baseline	MultiRes with normalization
IoU (%)	93.14	95.25
MCC (%)	94.91	96.74
Dice coefficient (%)	96.45	97.56

Comparative analysis between the baseline model and our proposed MultiRes with normalization.

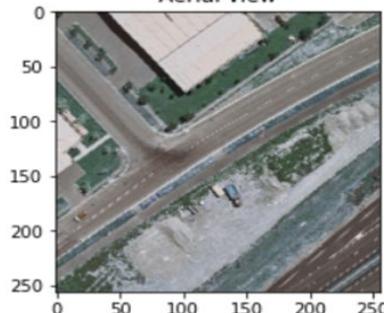
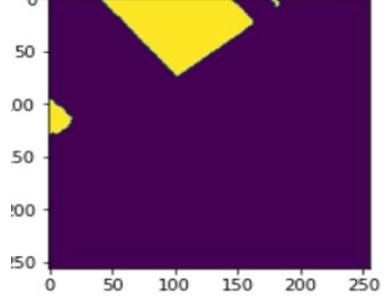
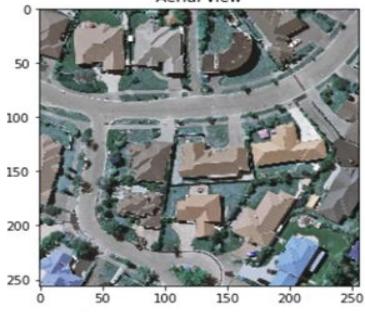
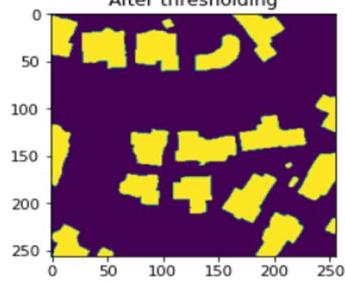
Model	Results on AIRS dataset			Results on Ankara, Turkey dataset		
	Precision	Recall	F1 score	Precision	Recall	F1 score
Shallow CNN	70	67.67	68	81	81.67	81
ResNet50	91.34	90.67	90.67	82.34	86.34	84.34
EfficientNetB4	91	89	89.67	81	84	82.34
VGG16	91	88.67	89.67	84.67	89.67	86.67

Comparative analysis of different DL models on AIRS dataset and dataset from Ankara, Turkey

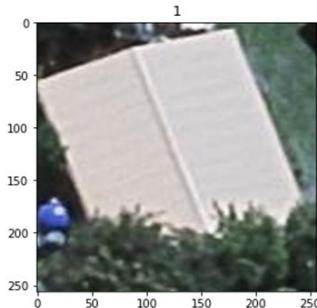
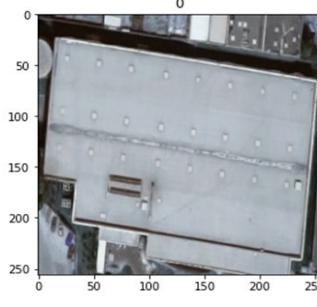
# TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_01	Applying Threshold	<p>Before thresholding</p> 	<p>After thresholding</p> 
TC_02	Building segmentation results- with no buildings.	<p>Aerial view</p> 	<p>After thresholding</p> 

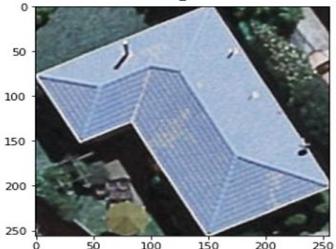
# TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_03	Building segmentation of a single building.	<p>Aerial view</p>  <p>0 50 100 150 200 250</p> <p>0 50 100 150 200 250</p>	<p>After thresholding</p>  <p>0 50 100 150 200 250</p> <p>0 50 100 150 200 250</p>
TC_04	Building segmentation of various buildings in a single image.	<p>Aerial view</p>  <p>0 50 100 150 200 250</p> <p>0 50 100 150 200 250</p>	<p>After thresholding</p>  <p>0 50 100 150 200 250</p> <p>0 50 100 150 200 250</p>

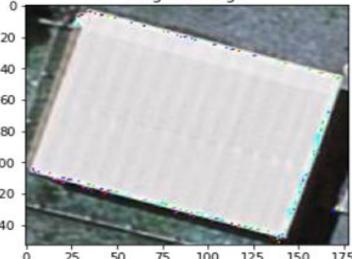
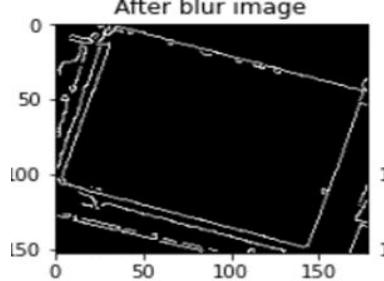
# TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_05	Classification of Gable image	 A photograph of a house roof with a gable end, labeled with a blue marker. The image is a 2D plot with axes ranging from 0 to 250.	ResNet50 - 1 (Gable) Shallow CNN - 1 (Gable) EfficientNetB4 - 1 (Gable) VGG16 - 1 (Gable)
TC_06	Classification of flat image	 A photograph of a flat roof with a grid pattern, labeled with a blue marker. The image is a 2D plot with axes ranging from 0 to 250.	ResNet50 - 0 (Flat) Shallow CNN - 1 (Gable) EfficientNetB4 - 0 (Flat) VGG16 - 0 (Flat)

# TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_07	Classification of hip image	 <p>Aerial photograph of a house with a complex, multi-peaked roof structure, labeled as a hip roof.</p>	<p>ResNet50 - 2 (Hip)          Shallow CNN - 2 (Hip)          EfficientNetB4 - 2 (Hip)          VGG16 - 2 (Hip)</p>
TC_08	Majority Voting (Unequal predictions)	 <p>Aerial photograph of a house with a gabled roof structure.</p>	<p>CNN Result: 2          ResNet Result: 1          EfficientNetB4-Adam Result: 1          VGG Result: 1          [[2 1 1 1]]          [0, 3, 1] 1          Image christchurch_173_343.jpg belongs to class: 1</p> <p>3 models predict as Gable while 1 model predicts as Flat</p>

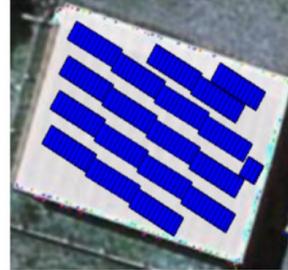
# TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_09	Majority Voting (Equal predictions)		<p>CNN Result: 2      ResNet Result: 2      EfficientNetB4-Adam Result: 2      VGG Result: 2      [[2 2 2 2]]      [0, 0, 4] 2      Image christchurch_173_395.jpg belongs to class: 2</p> <p>All models predict as class hip</p>
TC_10	Auto Canny Edge detection	 <p>Original image</p>	 <p>After blur image</p>

# TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_11	PV Panel placement for flat roof	Roof type: Flat Length: 20mm Width: 10 mm Tilt Angle: 40	
TC_12	PV Panel placement for flat roof with different width	Roof type: Flat Length: 20mm Width: 20 mm Tilt Angle: 40	

## TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_13	PV Panel placement for flat roof with varying tilt angle	Roof type: Flat Length: 20mm Width: 20 mm Tilt Angle: 30	
TC_14	PV Panel placement for gable roof with tree as obstacle	Roof type: Gable Length: 20mm Width: 20 mm Tilt Angle: 10	 <small>TEL = TEL,TYPE,TYPE(TEL / TELTYPE)</small>

## TEST CASES

TEST CASE ID	TEST CASE DESCRIPTION	TEST INPUT	TEST OUTPUT
TC_15	PV Panel placement for flat roof with tree as obstacle	Roof type: Flat Length: 10mm Width: 10 mm Tilt Angle: 30	

## CONCLUSION

- ★ A 3-step pipeline was proposed in this research to automate the process of PV panel placement on rooftops given a satellite image covering a wide range and different types of buildings.
- ★ The first phase involved building segmentation on AIRS dataset with the MultiRes UNet model that utilizes MultiRes block to adapt spatial features from various scales and Res path for transferring encoder features to decoder features using a set of convolution operations.
- ★ Experiment results shows that proper segmentation of buildings led to proper extraction of different roof types which were manually categorized into 3 classes: Flat, Gable and Hip.
- ★ We discovered that different models performed well in classifying different classes of roof tops with ResNet performing well in identifying flat roofs, EfficientNetB4 in classifying gable roof tops and VGG16 with hip roofs and hence majority voting was used as an ensembling technique in combining the predictions of different learning models.
- ★ The final output classifies the type of roof into one of flat, gable or hip and simulates the modular layout fitting of PV panels on top of roofs considering the type of roof, tilt angle, length and width of roof types.

## REFERENCES

- [1] Abdollahi, A., Pradhan, B., Gite, S., & Alamri, A. (2020). Building footprint extraction from high resolution aerial images using generative adversarial network (GAN) architecture. *IEEE Access*, Article 209517-209527. <http://dx.doi.org/10.1109/ACCESS.2020.3038225>.
- [2] Axelsson, M., Soderman, U., Berg, A. and Lithen, T., 2018, April. Roof Type Classification Using Deep Convolutional Neural Networks on Low Resolution Photogrammetric Point Clouds from Aerial Imagery. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)(pp.1293-1297).IEEE, <https://doi.org/10.1109/ICASSP.2018.846>.
- [3] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 39, 2481-2495.
- [4] B. Chatterjee and C. Poullis, "On Building Classification from Remote Sensor Imagery Using Deep Neural Networks and the Relation Between Classification and Reconstruction Accuracy Using Border Localization as Proxy," (2019) 16th Conference on Computer and Robot Vision (CRV), 2019, pp. 41-48, doi: 10.1109/CRV.2019.00014.
- [5] Buyukdemircioglu, Mehmet & Can, Recep & Kocaman, Sultan (2021)Deep learning based roof type classification using VHR aerial imagery, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. XLIII-B3-2021. 55-60. 10.5194/isprs-archives-XLIIIB3-2021-55-2021.
- [6] Chen, Mengge and Jonathan Li. (2019) “Deep convolutional neural network application on rooftop detection for aerial image.” ArXiv abs/1910.13509.
- [7] Edun, Ayobami & Harley, Joel & Deline, Chris & Perry, Kirsten. (2021). Unsupervised azimuth estimation of solar arrays in low-resolution satellite imagery through semantic segmentation and Hough transform. *Applied Energy*. 298. 10.1016/j.apenergy.2021.117273.

## REFERENCES

- [8] Jordan M Malof, Rui Hou, Leslie M Collins, Kyle Bradbury, and Richard Newell. 2015. Automatic solar photovoltaic panel detection in satellite imagery. In 2015 International Conference on Renewable Energy Research and Applications (ICRERA). IEEE, 1428-1431.
- [9] Kumar, Akash & Sreedevi, Indu. (2018). Solar Potential Analysis of Rooftops Using Satellite Imagery. ArXiv abs/1812.11606.
- [10] Nahid Mohajeri, Dan Assouline, Berenice Guiboud, Andreas Bill, Agust Gudmundsson, Jean-Louis Scartezzini, (2018). A city-scale roof shape classification using machine learning for solar energy applications, Renewable Energy Volume 121. Pages 81-93. ISSN 0960-1481. doi:10.1016/j.renene.2017.12.096.
- [11] Peiran Li, Haoran Zhang, Zhiling Guo, Suxing Lyu, Jinyu Chen, Wenjing Li, Xuan Song, Ryosuke Shibasaki, Jinyue Yan. (2021). Understanding rooftop PV panel semantic segmentation of satellite and aerial images for better using machine learning. Advances in Applied Energy, Elsevier. Volume 4, 100057, ISSN 2666-7924. doi: 10.1016/j.adapen.2021.100057.
- [12] Qi, Chen & Wang, Lei & Wu, Yifan & Wu, Guangming & Guo, Zhiling & Waslander, Steven. (2018). Aerial Imagery for Roof Segmentation: A Large-Scale Dataset towards Automatic Mapping of Buildings. ISPRS Journal of Photogrammetry and Remote Sensing, Elsevier. Volume 147, pp. 42-55.
- [13] Q. Li, Y. Feng, Y. Leng and D. Chen, " SolarFinder: Automatic Detection of Solar Photovoltaic Arrays," (2020) 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), 2020, pp. 193-204, doi: 10.1109/IPSN48710.2020.00024.
- [14] V. Golovko, S. Bezobrazov, A. Kroshchanka, A. Sachenko, M. Komar and A. Karachka, (2017), "Convolutional neural network based solar photovoltaic panel detection in satellite photos," 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), pp. 14-19, doi: 10.1109/IDAACS.2017.8094501.
- [15] X. Li, Y. Jiang, H. Peng and S. Yin, (2019), "An aerial image segmentation approach based on enhanced multi-scale convolutional neural network," IEEE International Conference on Industrial Cyber Physical Systems (ICPS), pp. 47-52, doi: 10.1109/ICPHYS.2019.8780187.

THANK YOU

	A	B	C	D	E	F	G	H	I	J	K	L
1	Model Name	Architecture	Numpy or Image Generator	Epochs	Batch size	Learning Rate	No of images	Performance Metrics	Colab or Kaggle	Filename	Comments	Results
2	u-net-1000images-baseline	U-Net All U-Net images are of dimensions 256 * 256	Numpy	10	8	0.01	1008	IOU, Accuracy	Colab	Basic U-Net	Scaling done by directly by 255	IOU - 18.06, Accuracy - 95.81
3	unet-youtube-1000images-10 epochs	U-Net	Numpy	10	16	0.01	1008	IOU, Dice Coefficient, Accuracy	Colab	Youtube U-Net	Used MinMaxScaler, Referred Digital Screen video	IOU - 64.73, Dice Coefficient - 78.43 Accuracy - 94.84
4	unetmcc-youtube-1000image s-10epochs	U-Net	Numpy	10	16	0.01	1008	IOU, Dice Coefficient, MCC, Accuracy	Colab	Youtube U-Net	Used MinMaxScaler, Added MCC also as performance metrics, Referred Digital Screen video	IOU - 68.22, Dice Coefficient - 80.99, MCC - 14.40, Accuracy - 95.37
5	resumable-model	U-Net	Numpy	15	8	0.001	1008	IOU, Dice Coefficient, MCC, Accuracy	Colab	Youtube U-Net	Same as above, Tried How to start, stop resume training of a model using keras-buoy	IOU - 70.36, Dice Coefficient - 82.34, MCC - NaN, Accuracy - 94.98
6	unet2-1000images-100epoch s	U-Net	Numpy	100	8	0.0001	1008	IOU, Dice Coefficient, MCC, Accuracy	Colab	U-Net Final	Tried resumable models too but that wasn't done properly	IOU - 88.40, Dice Coefficient - 93.79, MCC - 51.68, Accuracy - 98.22, Loss - 0.0356
7	unet2-1000images-130epoch s	U-Net	Numpy	30 (Continue training from the previous model) 100+30=130	8	0.00001	1008	IOU, Dice Coefficient, MCC, Accuracy	Colab	U-Net Final	Continued training from the previous model by loading the weights and decreasing the learning rate. Trained for 30 more epochs.	IOU - 90.57, Dice Coefficient - 95.02, MCC - 56.88, Accuracy - 98.48, Loss - 0.0279

	A	B	C	D	E	F	G	H	I	J	K	L
8	1500images-npy-100epochs	U-Net	Numpy	100	8	0.0001	1548	IOU, Dice Coefficient, MCC, Accuracy	Kaggle	1500 Numpy MultiRes Version 1 - Numpy MultiRes,U-Net	UNet with numpy array. Ran faster than in Colab. Took 48s for each epoch in Kaggle compared to 2mins in Colab.	IOU - 86.81, Dice Coefficient - 92.83, MCC - 48.97, Accuracy - 98.28, Loss - 0.0339
9	focalloss-30epochs	MultiRes	Numpy	100	8	0.0001	1008	IOU, Dice Coefficient, MCC, Accuracy	Kaggle	Keras Data Generator Version 1 - multires-30	MultiRes with binary focal loss instead of binary cross entropy loss.	IOU - 32.33, Dice Coefficient - 48.43, MCC - 0.88, Accuracy - 97.73, Loss - 0.0422
10	1500images-30epochs	MultiRes	Numpy	30 (256*192)	8	0.0001	1548	IOU, Dice Coefficient, MCC, Accuracy	Kaggle	1500 Numpy MultiRes Version 1 - Numpy MultiRes,U-Net	Tried MultiRes UNet with numpy array.	IOU - 30.21, Dice Coefficient - 45.87, MCC - 0.89, Accuracy - 97.53, Loss - 0.3288
11	multires-100epochs	MultiRes	Numpy	100 (256*192)	8	0.01	500	IOU, Dice Coefficient, MCC, Accuracy	Colab	MultiRes UNet v1	MultiRes UNet (original filter) not as in base paper. The output segmentation masks results seem somewhat good and performance has improved slightly.	IOU - 39.42, Dice Coefficient - 56.53, MCC - 2.77, Accuracy - 94.44, Loss - 0.2037
12	Dropout-500images-30epoch s-MultiRes	MultiRes	Numpy	30 (256*192)	8	0.01	500	IOU, Dice Coefficient, MCC, Accuracy	Kaggle	1500 Numpy MultiRes Version 3 - Dropout-MultiRes-500	Used Dropout(0.3) in conv_2d_bn function. 56 dropouts	IOU - 30.58, Dice Coefficient - 46.39, MCC - 2.07, Accuracy - 90.41, Loss - 0.3374



## POTSDAM SAMPLE IMAGES



Flat1



Flat2



Flat3



Flat4



Gable28



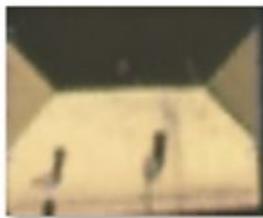
Gable29



Gable30



Gable31



Hip54



Hip55



Hip56



Hip57

Untitled spreadsheet

File Edit View Insert Format Data Tools Help Last edit was on April 10

A1 S.No. fx S.No.

	A	B	C	D
1	S.No.	Image Name	Type	Annotated or not
2	1	1.tif	Gable	Yes
3	2	3.tif	Complex	
4	3	6.tif	Flat	
5	4	8.tif	Gable	Yes
6	5	15.tif	Gable	To be deleted
7	6	16.tif	Gable	Yes
8	7	18.tif	Flat	
9	8	27.tif	Flat	
10	9	29.tif	Gable	Yes
11	10	30.tif	Gable	Yes
12	11	35.tif	Gable	Yes
13	12	christchurch_36_3.tif	Hip	
14	13	christchurch_36_5.tif	Gable	Yes
15	14	christchurch_36_6.tif	Gable	Yes
16	15	christchurch_36_16.tif	Complex	
17	16	christchurch_36_18.tif	Complex	
18	17	christchurch_36_29.tif	Flat	
19	18	christchurch_36_30.tif	Gable	Yes
20	19	christchurch_36_31.tif	Gable	Yes
21	20	christchurch_36_33.tif	Gable	Yes
22	21	christchurch_36_37.tif	Complex	

+ Rooftop\_Types

Untitled spreadsheet

File Edit View Insert Format Data Tools Help Last edit was on April 10

A1 fx S.No.

	A	B	C	D
1348	1347	yeyyO-31924	Flat	
1349	1348	yfryl-86650	Flat	
1350	1349	YFWSK-53068	Flat	
1351	1350	YKzLz-39951	Flat	
1352	1351	yIBSJ-55543	Flat	
1353	1352	YIJCc-59362	Flat	
1354	1353	YmXQu-43633	Flat	
1355	1354	ynpbI-89067	Flat	
1356	1355	YoQof-93452	Flat	
1357	1356	ypBoe-13811	Flat	
1358	1357	ySYoW-48498	Flat	
1359	1358	yvcqD-32337	Flat	
1360	1359	yVFhd-99563	Flat	
1361	1360	yWZlF-99489	Flat	
1362	1361	YyKlj-24618	Flat	
1363	1362	yZZlt-35376	Flat	
1364	1363	zawos-61611	Flat	
1365	1364	ZNrwf-45215	Flat	
1366	1365	znsrP-97157	Flat	
1367	1366	ZorrT-16821	Flat	
1368	1367	ZQQrC-78932	Flat	
1369	1368	ztbuf-94418	Flat	

+ Rooftop\_Types

A1	Model Name	Dataset	Architecture	Epochs	Learning Rate	Optimizer	Batch Size	Performance Metrics	Filename	Comments	Results	Resu
2	./ResNet50-finetuning-bs-8	Modified dataset: roottype-dataset-mod Flat - 374 Gable - 429 Hip - 319	ResNet50	100	0.00001	RMSProp	8	Accuracy, Classification Report	Classification with mod data - Version 0 - resnet & efficientnet	Gable misclassified as flat in potsdam dataset	Loss: 0.0871 Accuracy: 0.9753 Val Loss: 0.3376 Val Accuracy: 0.9815 Test Loss: 0.7633 Test Accuracy: 0.8667 F1-score: 0.86 Recall: 0.86 Precision: 0.86	Loss: Accur: 300 ir class
3	./EfficientNetB4-finetuning-bs16-RMS	Modified dataset: roottype-dataset-mod Flat - 374 Gable - 429 Hip - 319	EfficientNetB4	100	0.00001	RMSProp	16	Accuracy, Classification Report	Classification with mod data - Version 0 - resnet & efficientnet	Gable misclassified as flat in potsdam dataset	Loss: 0.3233 Accuracy: 0.8756 Val Loss: 0.2548 Val Accuracy: 0.92593 Test Loss: 0.3158 Test Accuracy: 0.8889 F1-score: 0.89 Recall: 0.89 Precision: 0.89	Loss: Accur: 300 ir class
4	./EfficientNetB4-finetuning-	Modified dataset: roottype-dataset-mod						Accuracy,	Classification with mod data - Version 1 - EfficientNet Adam-		Loss: 0.1280 Accuracy: 0.9552 Val Loss: 0.1322 Val Accuracy: 0.9815 Test Loss: 0.3158 Test Accuracy: 0.8889 F1-score: 0.91 Recall: 0.91	Loss: Accur: 300 ir

A1	A	B	C	D	E	F	G	H	I	J	K	
5	./VGG16-finetuning-RMS-b s4	Modified dataset: roottype-dataset-mod Flat - 374 Gable - 429 Hip - 320	VGG16	100	0.00001	RMSProp	4	Accuracy, Classification Report, AU-ROC	Classification with mod data - Version 2 - VGG16 with ROC	Added Au-ROC metrics. Results are comparatively not so good for PotsDam testing	Loss: 0.1011 Accuracy: 0.9705 Val Loss: 1.0823 Val Accuracy: 0.9259 Test Loss: 1.5621 Test Accuracy: 0.8444 F1-score: 0.84 Recall: 0.85 Precision: 0.84 ROC_AUC score for 3 models: [0: 0.9519230769230769, 1: 0.9453781512605042, 2: 0.9311111111111111]	Loss: Accur: 300 ir class
6	./VGG16-finetuning-RMS-b s4	Modified dataset: roottype-dataset-mod Flat - 374 Gable - 429 Hip - 321	VGG16	100	0.00001	RMSProp	4	Accuracy, Classification Report, AU-ROC	Classification with mod data - Version 3 - TTA for VGG16	TTA has improved accuracy from 57.33 to 59	Loss: 0.1011 Accuracy: 0.9705 Val Loss: 1.0823 Val Accuracy: 0.9259 Test Loss: 1.5621 Test Accuracy: 0.8444 F1-score: 0.84 Recall: 0.85 Precision: 0.84 ROC_AUC score for 3 models: [0: 0.9519230769230769, 1: 0.9453781512605042, 2: 0.9311111111111111]	Loss: Accur: 300 ir class

A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	
8	./CNN-RMSProp-50epochs	Modified dataset: rooftype-dataset-mod Flat - 374 Gable - 429 Hip - 321	Shallow CNN	50	0.0001	RMSProp	16	Accuracy, Classification Report, AU-ROC	Classification with mod data - Version 5 - all models 50 epochs	Nearly all gable images are classified as hip.	Loss: 0.7179 Accuracy: 0.6736 Val Loss: 0.6047 Val Accuracy: 0.8148 Test Loss: 0.8446 Test Accuracy: 0.7111 F1-score: 0.71 Recall: 0.73 Precision: 0.72 ROC_AUC score for 3 models: [0: 0.9716577540106951, 1: 0.7470355731225298, 2: 0.856060606060606061]
9	./ResNet50-50epochs-bs8	Modified dataset: rooftype-dataset-mod Flat - 374 Gable - 429 Hip - 322	ResNet50	50	0.00001	RMSProp	8	Accuracy, Classification Report, AU-ROC	Classification with mod data - Version 5 - all models 50 epochs	Some gable are misclassified as flat. Better results than 100 epochs.	Loss: 0.1698 Accuracy: 0.9427 Val Loss: 0.2077 Val Accuracy: 0.9444 Test Loss: 0.8303 Test Accuracy: 0.8444 F1-score: 0.84 Recall: 0.85 Precision: 0.84 ROC_AUC score for 3 models: [0: 0.8689839572192514, 1: 0.91699604743083, 2: 1.0]

A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1	
10	./EfficientNetB4-50epochs-bs16-RMS	Modified dataset: rooftype-dataset-mod Flat - 374 Gable - 429 Hip - 323	EfficientNetB4	50	0.00001	RMSProp	8	Accuracy, Classification Report, AU-ROC	Classification with mod data - Version 5 - all models 50 epochs	Some gable are misclassified as flat. Better results than 100 epochs.	Loss: 0.4292 Accuracy: 0.8189 Val Loss: 0.2895 Val Accuracy: 0.9074 Test Loss: 0.7727 Test Accuracy: 0.6889 F1-score: 0.70 Recall: 0.72 Precision: 0.79 ROC_AUC score for 3 models: [0: 0.8235294117647058, 1: 0.83399209486166, 2: 0.9848484848484848]
11	./EfficientNetB4-50epochs-bs16-Adam	Modified dataset: rooftype-dataset-mod Flat - 374 Gable - 429 Hip - 324	EfficientNetB4	50	0.00001	Adam	8	Accuracy, Classification Report, AU-ROC	Classification with mod data - Version 5 - all models 50 epochs	Flat very accurately classified. Gable misclassified as flat. Better results than 50 epochs.	Loss: 0.3815 Accuracy: 0.8567 Val Loss: 0.3477 Val Accuracy: 0.8889 Test Loss: 0.5437 Test Accuracy: 0.8222 F1-score: 0.82 Recall: 0.82 Precision: 0.83 ROC_AUC score for 3 models: [0: 0.8957219251336899, 1: 0.9011857707509882, 2: 0.98737373737375]

