

Figure S1: The ReLU activation function

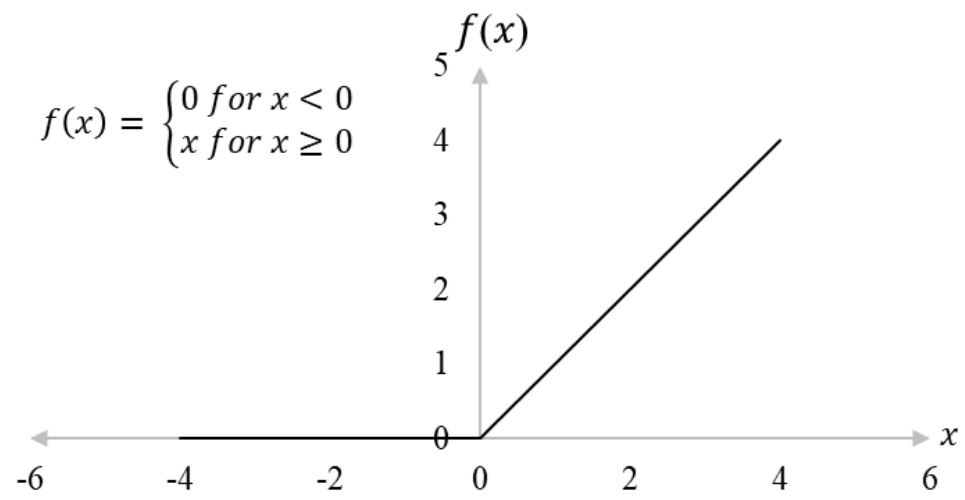


Figure S2: Plot used to choose the learning rate

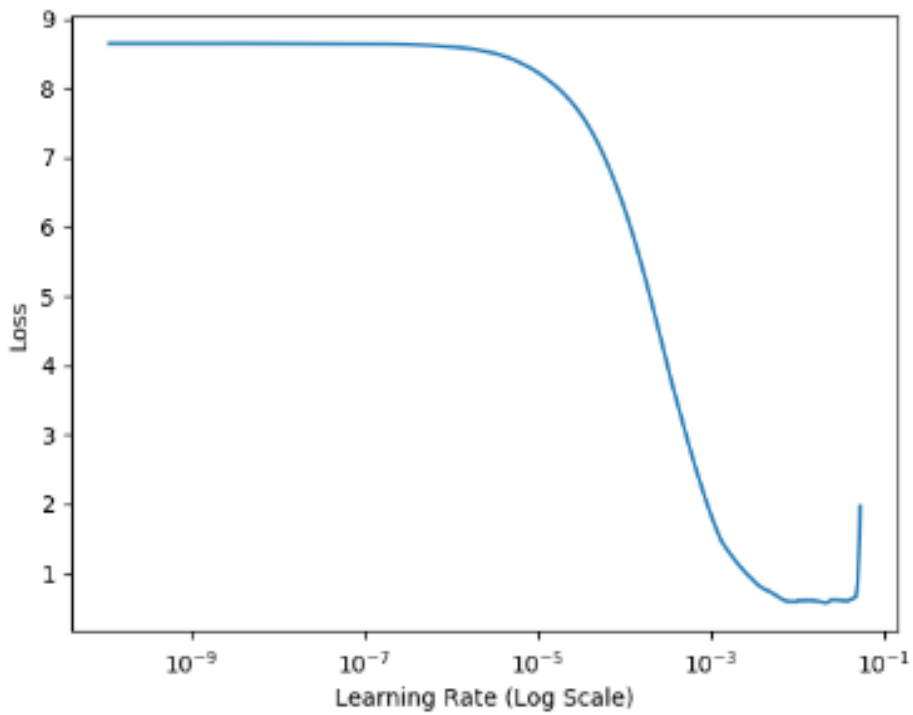


Figure S3: Subset of image and building mask used for training the model

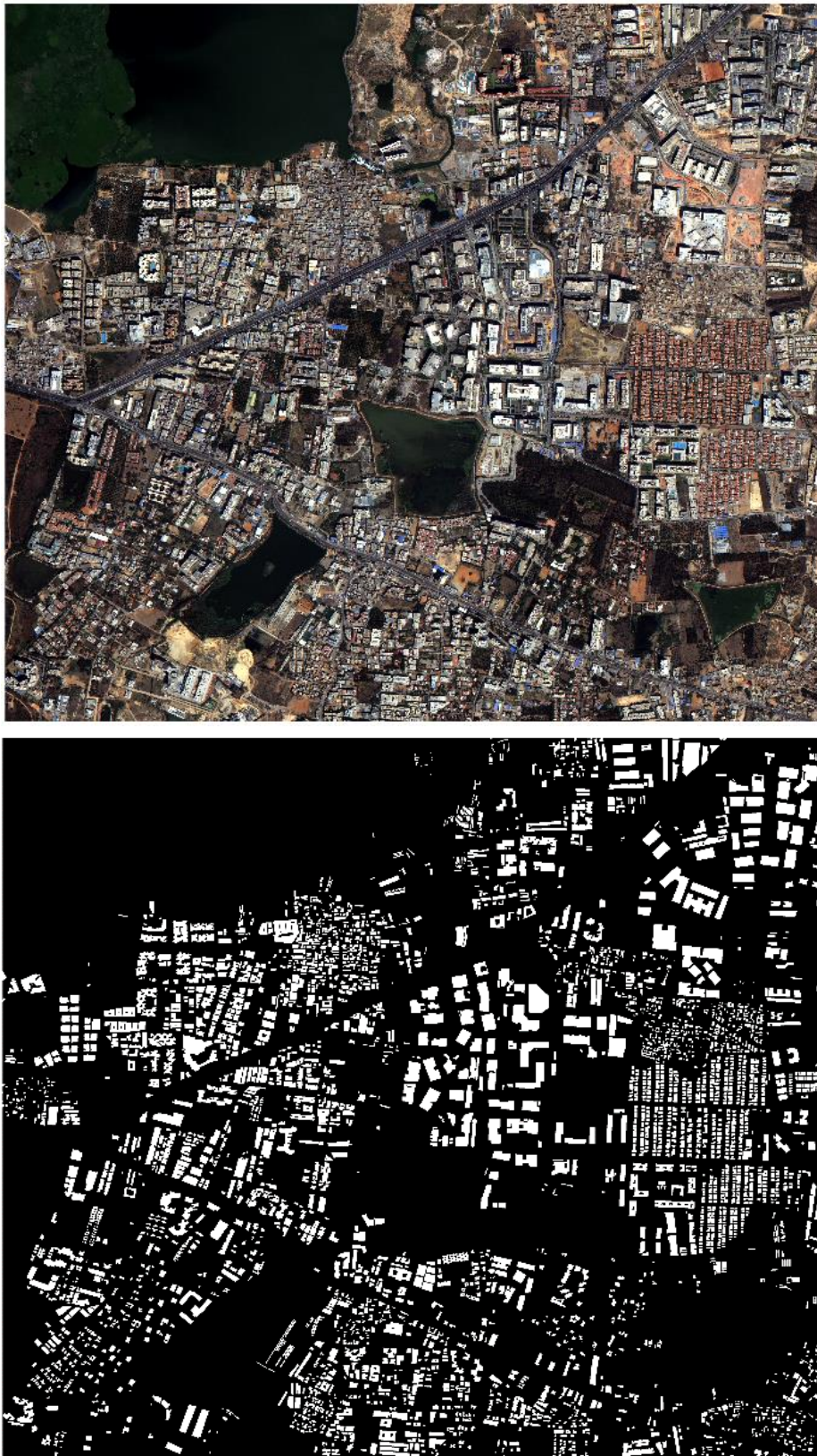


Figure S4: Model accuracy (left) and model loss (right) during the training process

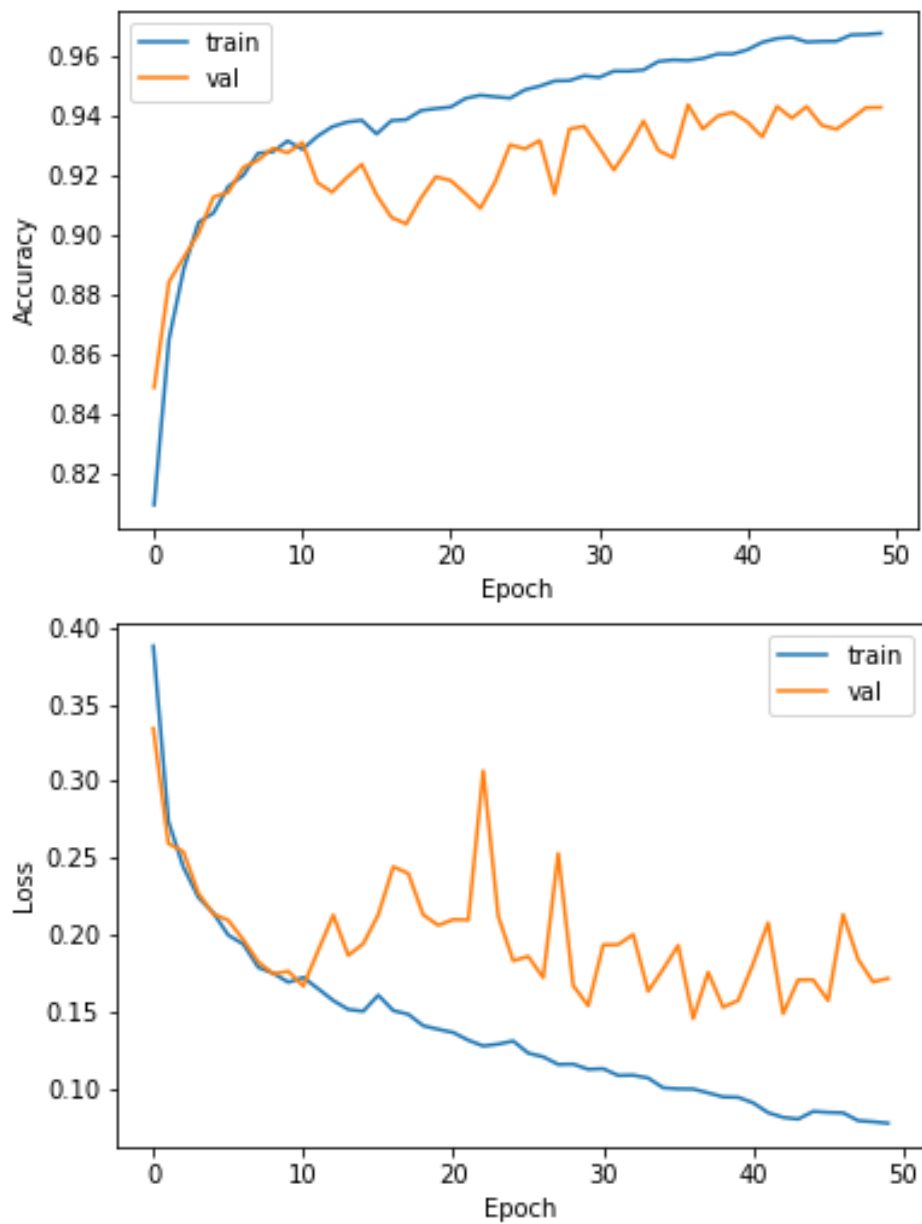
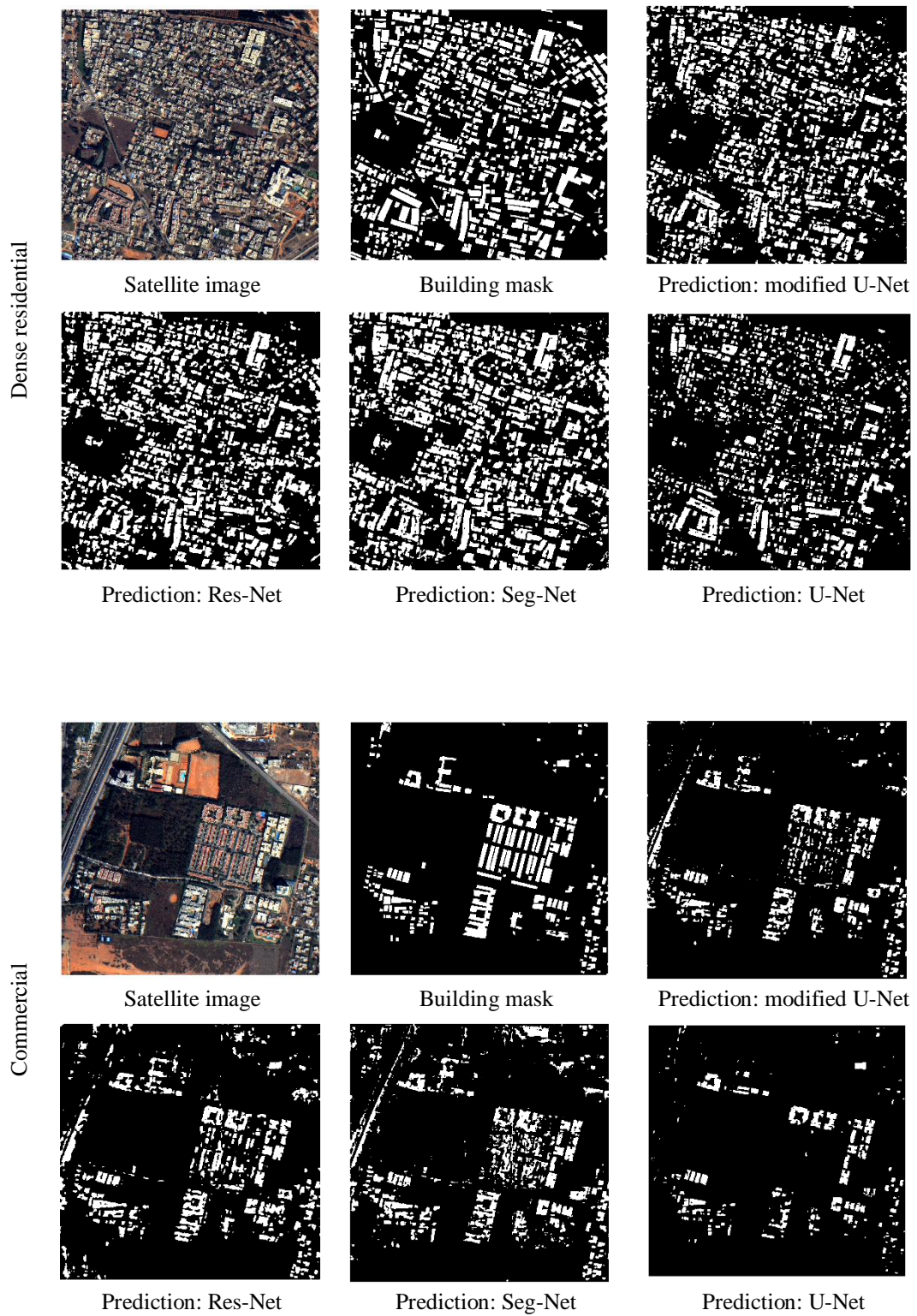


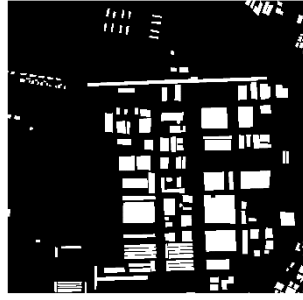
Figure S5: Model performance analysis over various built-up conditions



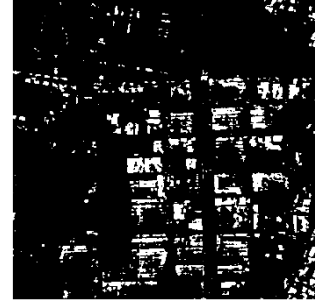
Industrial



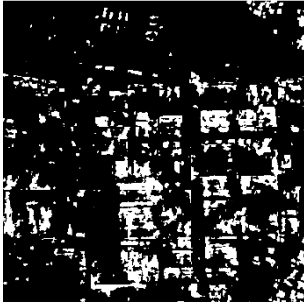
Satellite image



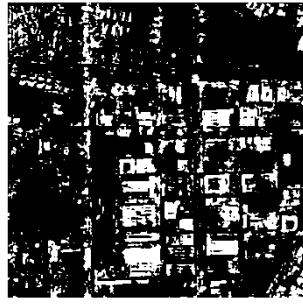
Building mask



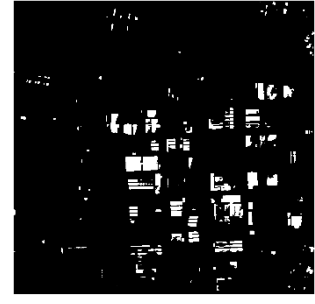
Prediction: modified U-Net



Prediction: Res-Net



Prediction: Seg-Net

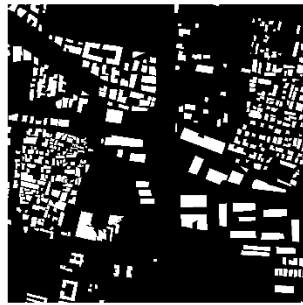


Prediction: U-Net

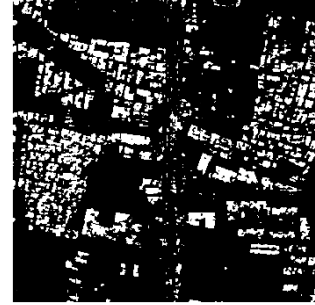
Mixed urban



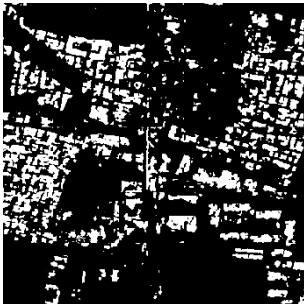
Satellite image



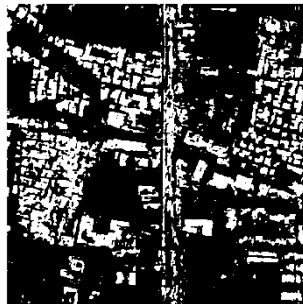
Building mask



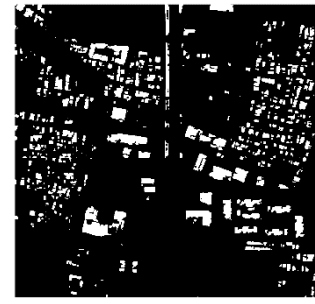
Prediction: modified U-Net



Prediction: Res-Net



Prediction: Seg-Net



Prediction: U-Net

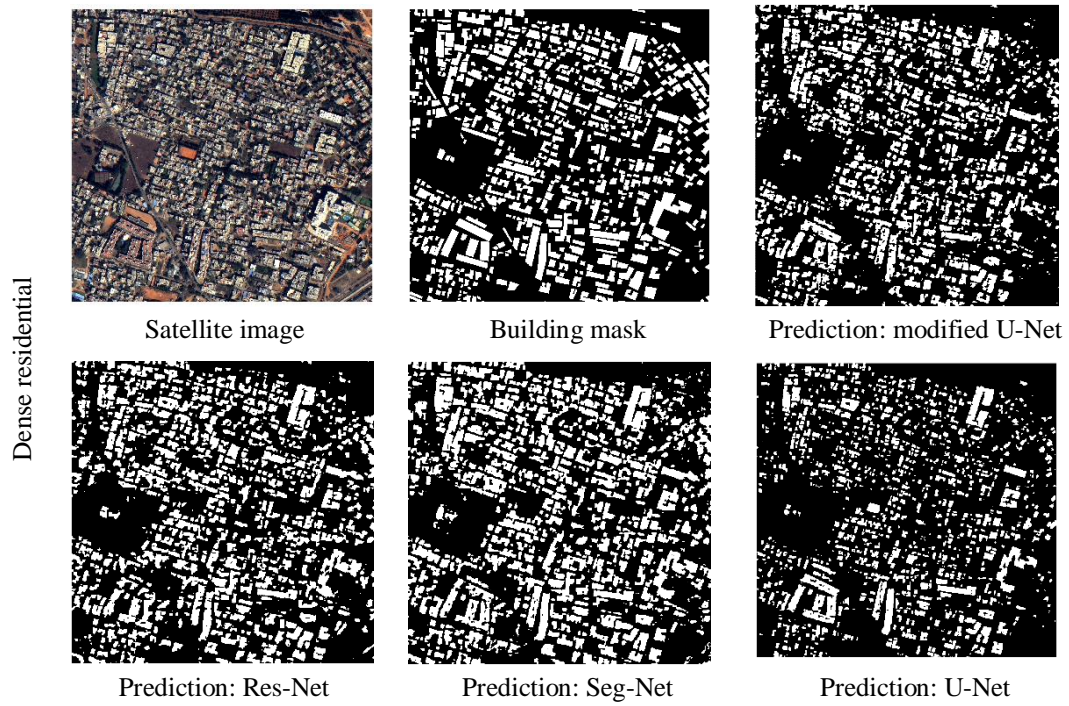
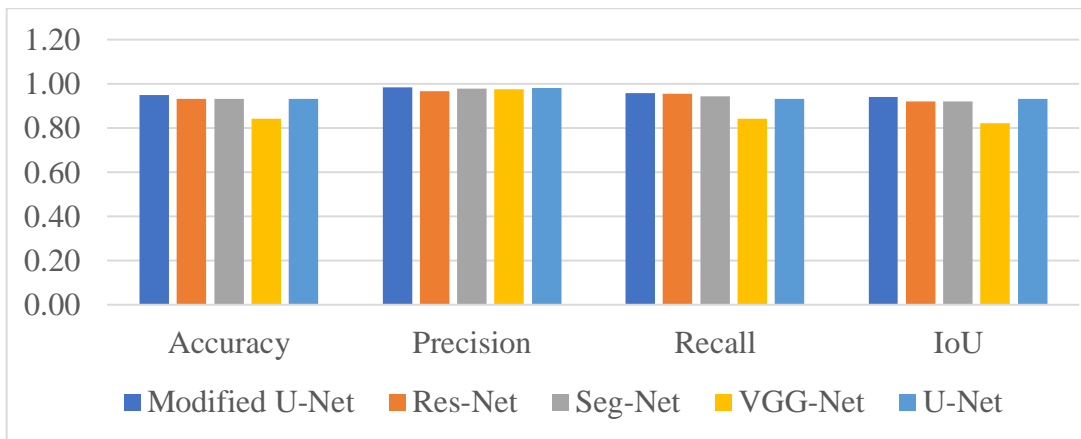


Figure S6: Performance of various models in sparse residential built-up condition



Appendix 1: Model summary (For image size 96×96)

Layer (type)	Output Shape	Parameters	Connected to
input_1 (InputLayer)	(None, 96, 96, 3)	0	
conv2d_1 (Conv2D)	(None, 96, 96, 32)	896	input_1[0][0]
conv2d_2 (Conv2D)	(None, 96, 96, 32)	9248	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 48, 48, 32)	0	conv2d_2[0][0]
batch_normalization_1 (BatchNor)	(None, 48, 48, 32)	128	max_pooling2d_1[0][0]
conv2d_3 (Conv2D)	(None, 48, 48, 64)	18496	batch_normalization_1[0][0]
conv2d_4 (Conv2D)	(None, 48, 48, 64)	36928	conv2d_3[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 24, 24, 64)	0	conv2d_4[0][0]
dropout_1 (Dropout)	(None, 24, 24, 64)	0	max_pooling2d_2[0][0]
batch_normalization_2 (BatchNor)	(None, 24, 24, 64)	256	dropout_1[0][0]
conv2d_5 (Conv2D)	(None, 24, 24, 128)	73856	batch_normalization_2[0][0]
conv2d_6 (Conv2D)	(None, 24, 24, 128)	147584	conv2d_5[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 12, 12, 128)	0	conv2d_6[0][0]
dropout_2 (Dropout)	(None, 12, 12, 128)	0	max_pooling2d_3[0][0]
batch_normalization_3 (BatchNor)	(None, 12, 12, 128)	512	dropout_2[0][0]
conv2d_7 (Conv2D)	(None, 12, 12, 256)	295168	batch_normalization_3[0][0]
conv2d_8 (Conv2D)	(None, 12, 12, 256)	590080	conv2d_7[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 256)	0	conv2d_8[0][0]
dropout_3 (Dropout)	(None, 6, 6, 256)	0	max_pooling2d_4[0][0]
batch_normalization_4 (BatchNor)	(None, 6, 6, 256)	1024	dropout_3[0][0]
conv2d_9 (Conv2D)	(None, 6, 6, 512)	1180160	batch_normalization_4[0][0]
conv2d_10 (Conv2D)	(None, 6, 6, 512)	2359808	conv2d_9[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 3, 3, 512)	0	conv2d_10[0][0]

dropout_4 (Dropout)	(None, 3, 3, 512)	0	max_pooling2d_5[0][0]
conv2d_11 (Conv2D)	(None, 3, 3, 1024)	4719616	dropout_4[0][0]
conv2d_12 (Conv2D)	(None, 3, 3, 1024)	9438208	conv2d_11[0][0]
conv2d_transpose_1 (Conv2DTrans)	(None, 6, 6, 512)	2097664	conv2d_12[0][0]
concatenate_1 (Concatenate) conv2d_10[0][0]	(None, 6, 6, 1024)	0	conv2d_transpose_1[0][0],
batch_normalization_5 (BatchNor)	(None, 6, 6, 1024)	4096	concatenate_1[0][0]
conv2d_13 (Conv2D)	(None, 6, 6, 512)	4719104	batch_normalization_5[0][0]
conv2d_14 (Conv2D)	(None, 6, 6, 512)	2359808	conv2d_13[0][0]
dropout_5 (Dropout)	(None, 6, 6, 512)	0	conv2d_14[0][0]
conv2d_transpose_2 (Conv2DTrans)	(None, 12, 12, 256)	524544	dropout_5[0][0]
concatenate_2 (Concatenate) conv2d_8[0][0]	(None, 12, 12, 512)	0	conv2d_transpose_2[0][0],
batch_normalization_6 (BatchNor)	(None, 12, 12, 512)	2048	concatenate_2[0][0]
conv2d_15 (Conv2D)	(None, 12, 12, 256)	1179904	batch_normalization_6[0][0]
conv2d_16 (Conv2D)	(None, 12, 12, 256)	590080	conv2d_15[0][0]
dropout_6 (Dropout)	(None, 12, 12, 256)	0	conv2d_16[0][0]
conv2d_transpose_3 (Conv2DTrans)	(None, 24, 24, 128)	131200	dropout_6[0][0]
concatenate_3 (Concatenate) conv2d_6[0][0]	(None, 24, 24, 256)	0	conv2d_transpose_3[0][0],
batch_normalization_7 (BatchNor)	(None, 24, 24, 256)	1024	concatenate_3[0][0]
conv2d_17 (Conv2D)	(None, 24, 24, 128)	295040	batch_normalization_7[0][0]
conv2d_18 (Conv2D)	(None, 24, 24, 128)	147584	conv2d_17[0][0]
dropout_7 (Dropout)	(None, 24, 24, 128)	0	conv2d_18[0][0]
conv2d_transpose_4 (Conv2DTrans)	(None, 48, 48, 64)	32832	dropout_7[0][0]
concatenate_4 (Concatenate) conv2d_4[0][0]	(None, 48, 48, 128)	0	conv2d_transpose_4[0][0],

batch_normalization_8 (BatchNor)	(None, 48, 48, 128)	512	concatenate_4[0][0]
conv2d_19 (Conv2D)	(None, 48, 48, 64)	73792	batch_normalization_8[0][0]
conv2d_20 (Conv2D)	(None, 48, 48, 64)	36928	conv2d_19[0][0]
dropout_8 (Dropout)	(None, 48, 48, 64)	0	conv2d_20[0][0]
conv2d_transpose_5 (Conv2DTrans)	(None, 96, 96, 32)	8224	dropout_8[0][0]
concatenate_5 (Concatenate) conv2d_2[0][0]	(None, 96, 96, 64)	0	conv2d_transpose_5[0][0], conv2d_2[0][0]
conv2d_21 (Conv2D)	(None, 96, 96, 32)	18464	concatenate_5[0][0]
conv2d_22 (Conv2D)	(None, 96, 96, 32)	9248	conv2d_21[0][0]
conv2d_23 (Conv2D)	(None, 96, 96, 1)	33	conv2d_22[0][0]
=====			
Total params: 31,104,097			
Trainable params: 31,099,297			
Non-trainable params: 4,800			

Appendix II

The GitHub link contains the python code for the complete deep learning implementation pipeline: https://github.com/prakashps/building_extraction