SCENE CLASSIFICATION

- Boyuan Du, Jiaoyue Sun, Saurav Mainali

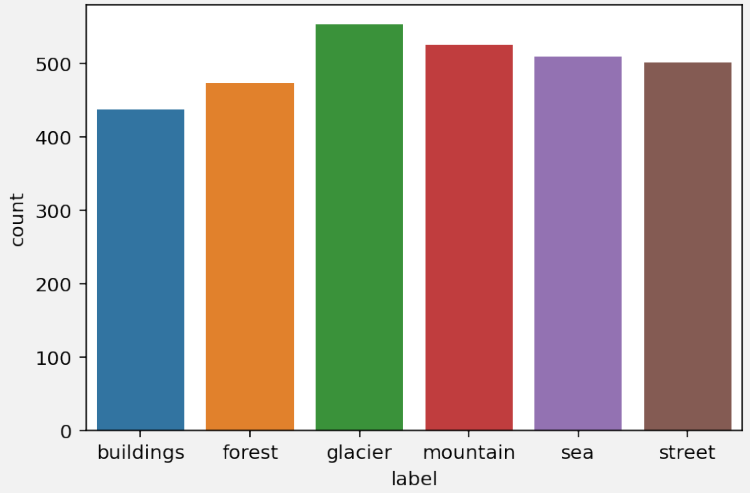
PROJECT DESCRIPTION

- **Problem :** Image Classification
- **Objective** : Classify Scenes around the world into one of its possible classes
- **Applications of Scene Classification** : Organization of photos in Smartphone ; Drive Economy through Tourism Planning and so on.

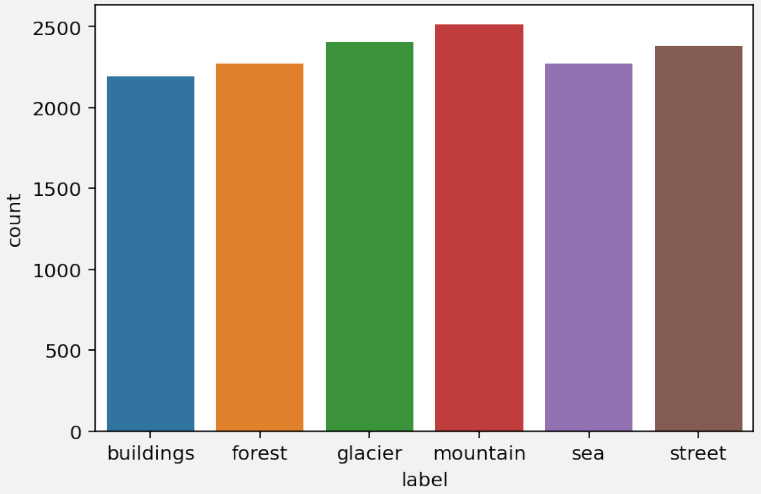
DATASET DESCRIPTION

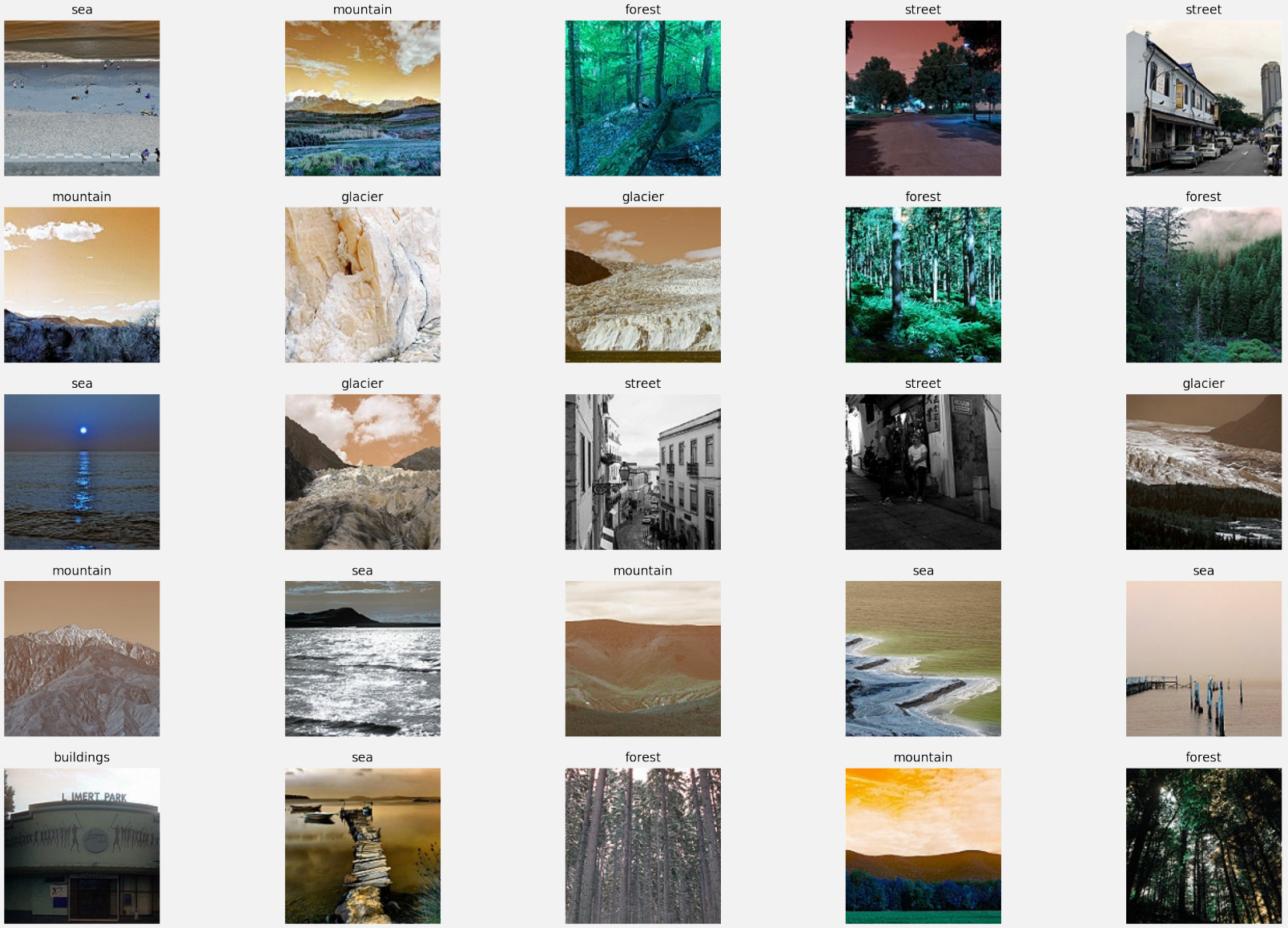
- **Source :** https://www.kaggle.com/puneet6060/intel-image-classification/download
- **Images** : Total of 25,000 images captured by Jan Bottinger; Out of which only 17,000 were labelled
- **Class Labels** : Buildings, Forest, Glacier, Mountain, Sea or Street
- **Data Split** : 70% Training ; 15% CV ; 15% Testing

Trainset



Testset





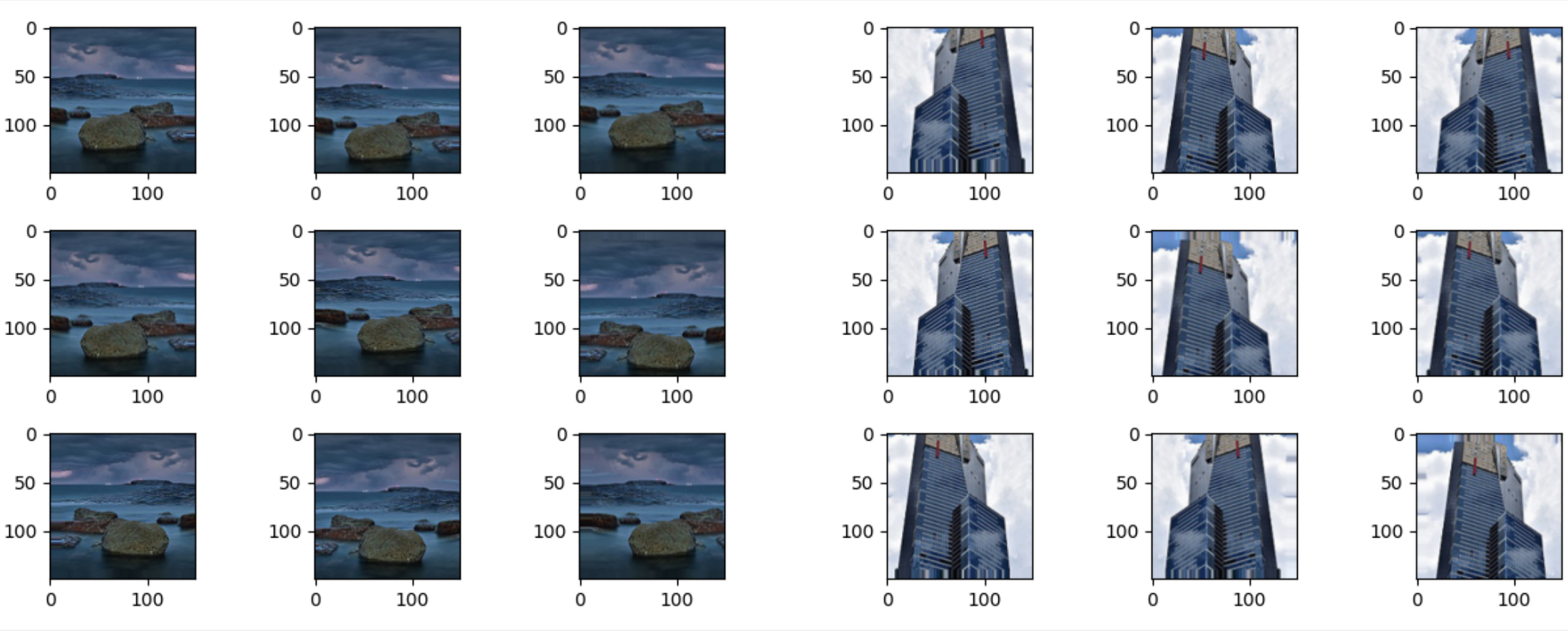
DATA PREPARATION

- Images were read using CV2 library to obtain matrix form of it
- They were then converted into 150 by 150 pixels
- Images were then finally normalized such that each pixel's value ranged from 0 to 1

PRETRAINED NETWORKS

- VGG16, VGG19, ResNet152V2, InceptionV3, InceptionResV2 were used form keras.applications
- Weights obtained by Training on ImageNet Dataset were used
- Updating these weights along with the later added layers dropped the performance of Pre-Trained Networks; so we fixed it

DATA AUGMENTATION



DATA_AUG RESULTS

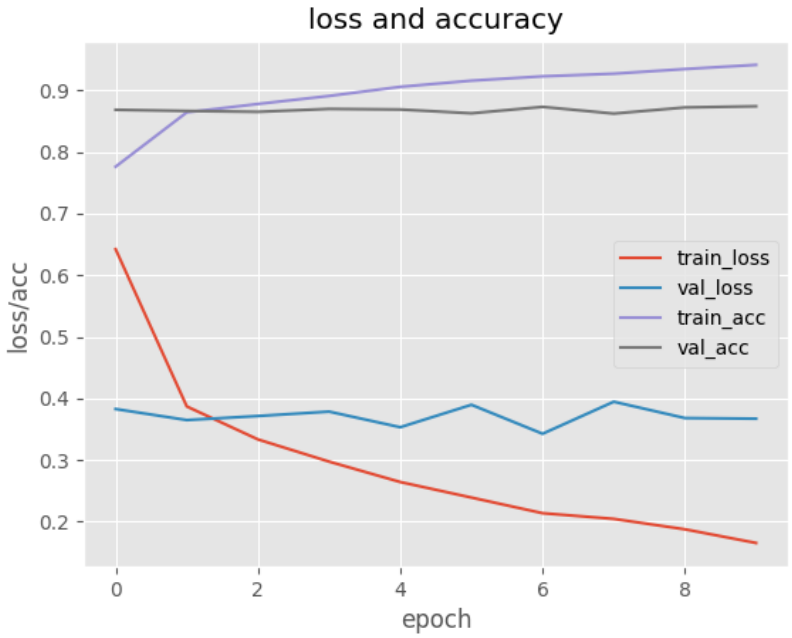
Model	Accuracy	Cohen Kappa Score	Macro F1 Score
VGG16	89.43 %	0.87	0.90
VGG19	88.73 %	0.86	0.89
ResNet152V2	87.27 %	0.84	0.88
InceptionV3	84.20 %	0.81	0.85
InceptionResV2	83.99 %	0.81	0.85

SELF-TRAINED CNN NETWORK

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 200)     5600
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 146, 146, 180)     324180
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 29, 29, 180)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 27, 27, 180)       291780
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 25, 25, 140)       226940
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 23, 23, 100)       126100
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 21, 21, 50)        45050
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 50)          0
_________________________________________________________________
flatten (Flatten)            (None, 800)               0
_________________________________________________________________
dense (Dense)                (None, 180)               144180
_________________________________________________________________
dense_1 (Dense)              (None, 100)               18100
_________________________________________________________________
dense_2 (Dense)              (None, 50)                5050
_________________________________________________________________
dropout (Dropout)            (None, 50)                0
_________________________________________________________________
dense_3 (Dense)              (None, 6)                 306
=================================================================
Total params: 1,187,286
Trainable params: 1,187,286
```

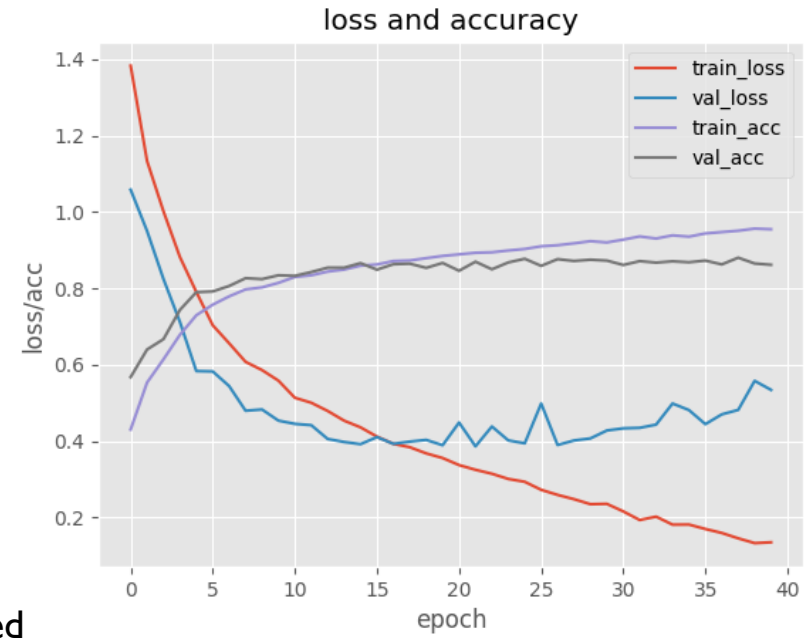
RESULTS

Model	Accuracy	Cohen Kappa Score	Macro F1 Score
VGG16	89.70 %	0.87	0.89
VGG19	88.49 %	0.86	0.88
ResNet152V2	90.63 %	0.88	0.90
InceptionV3	87.23 %	0.84	0.87
InceptionResV2	88.13 %	0.85	0.88
Self-Trained Model	86.09 %	0.77	0.83



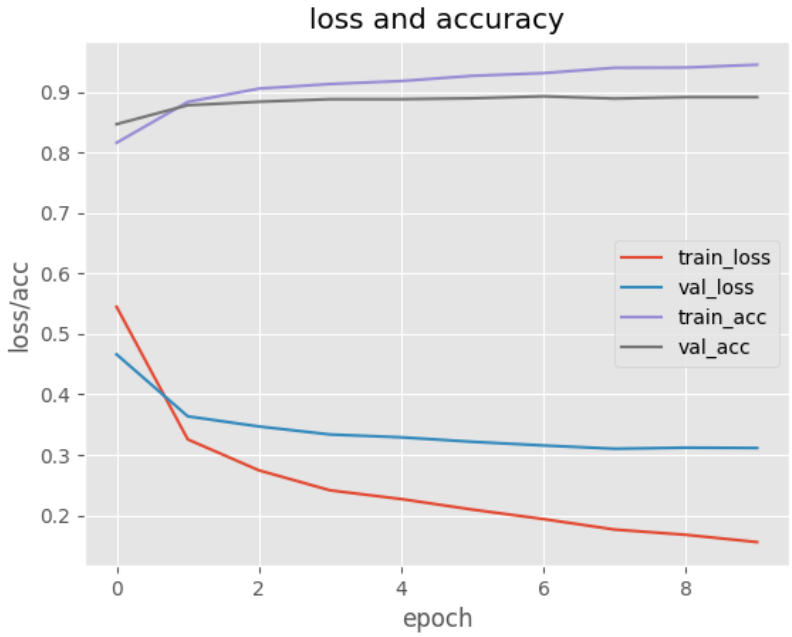
Vgg19

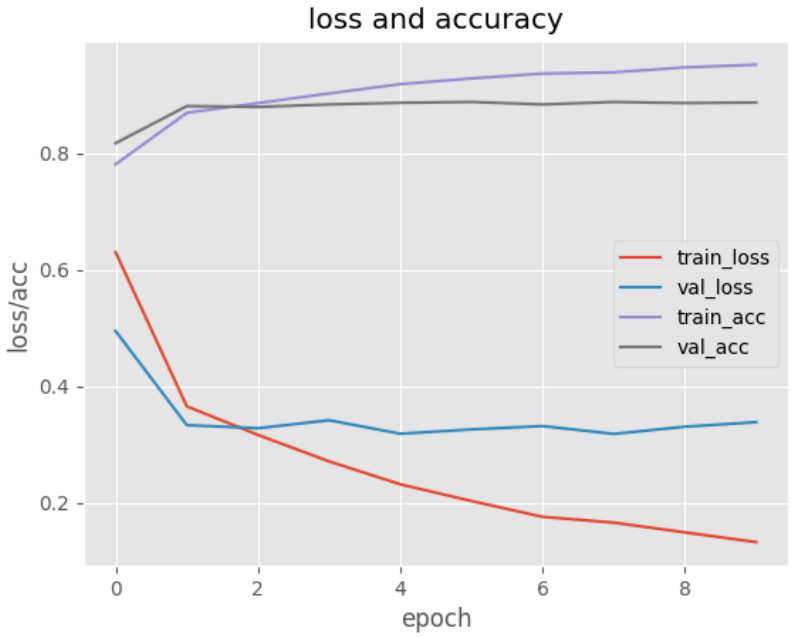
Self-Trained



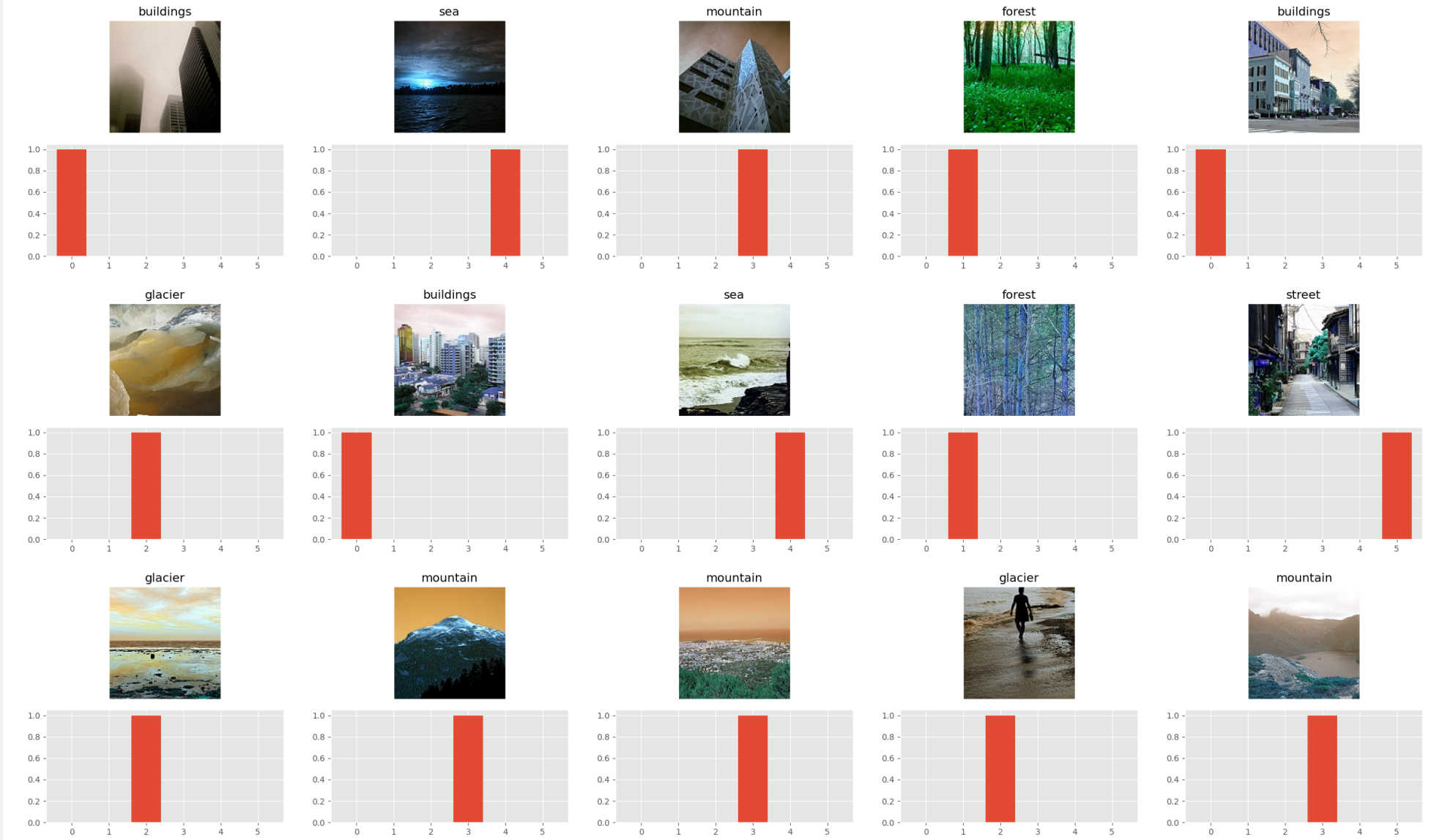
ResNet152V2

Vgg16

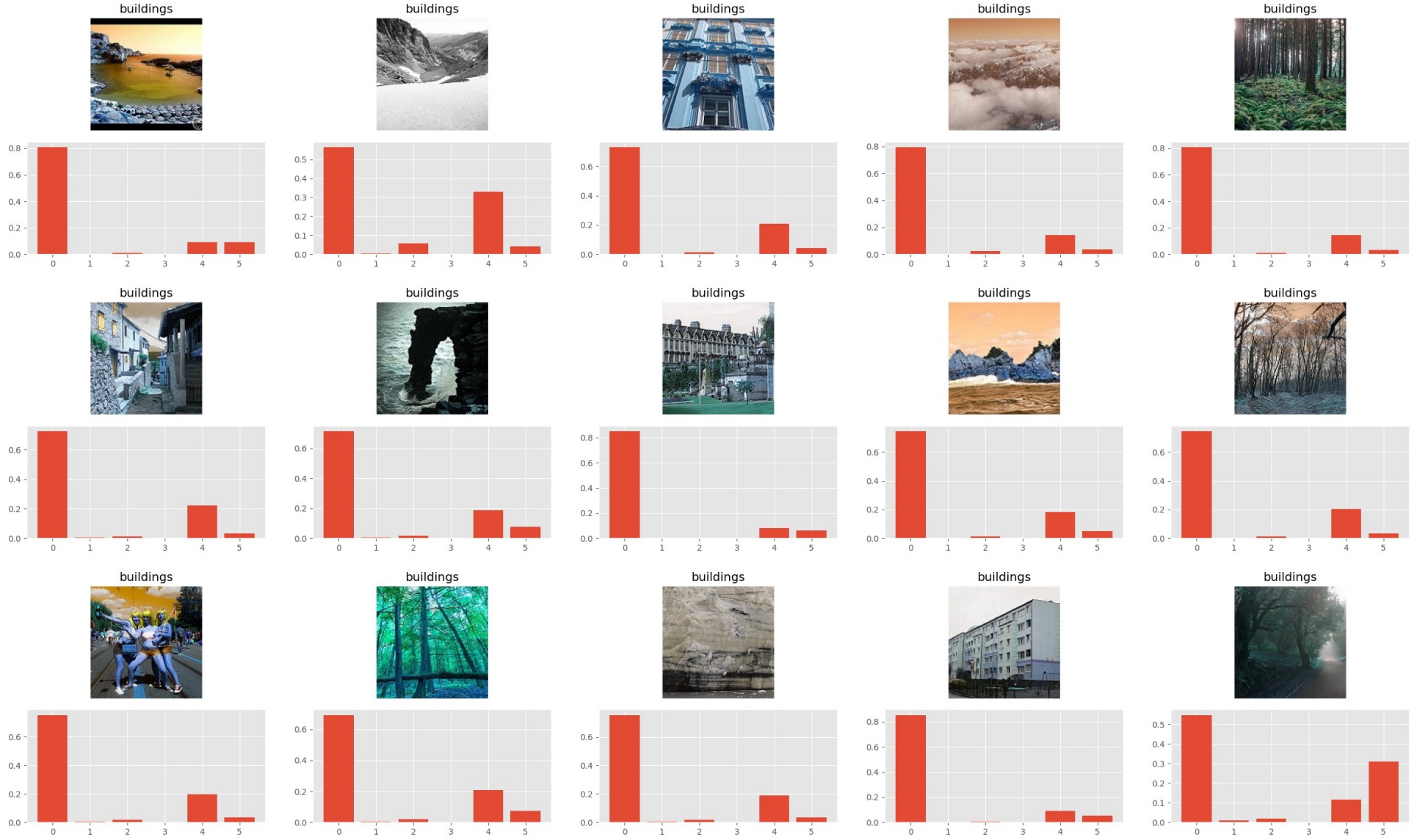




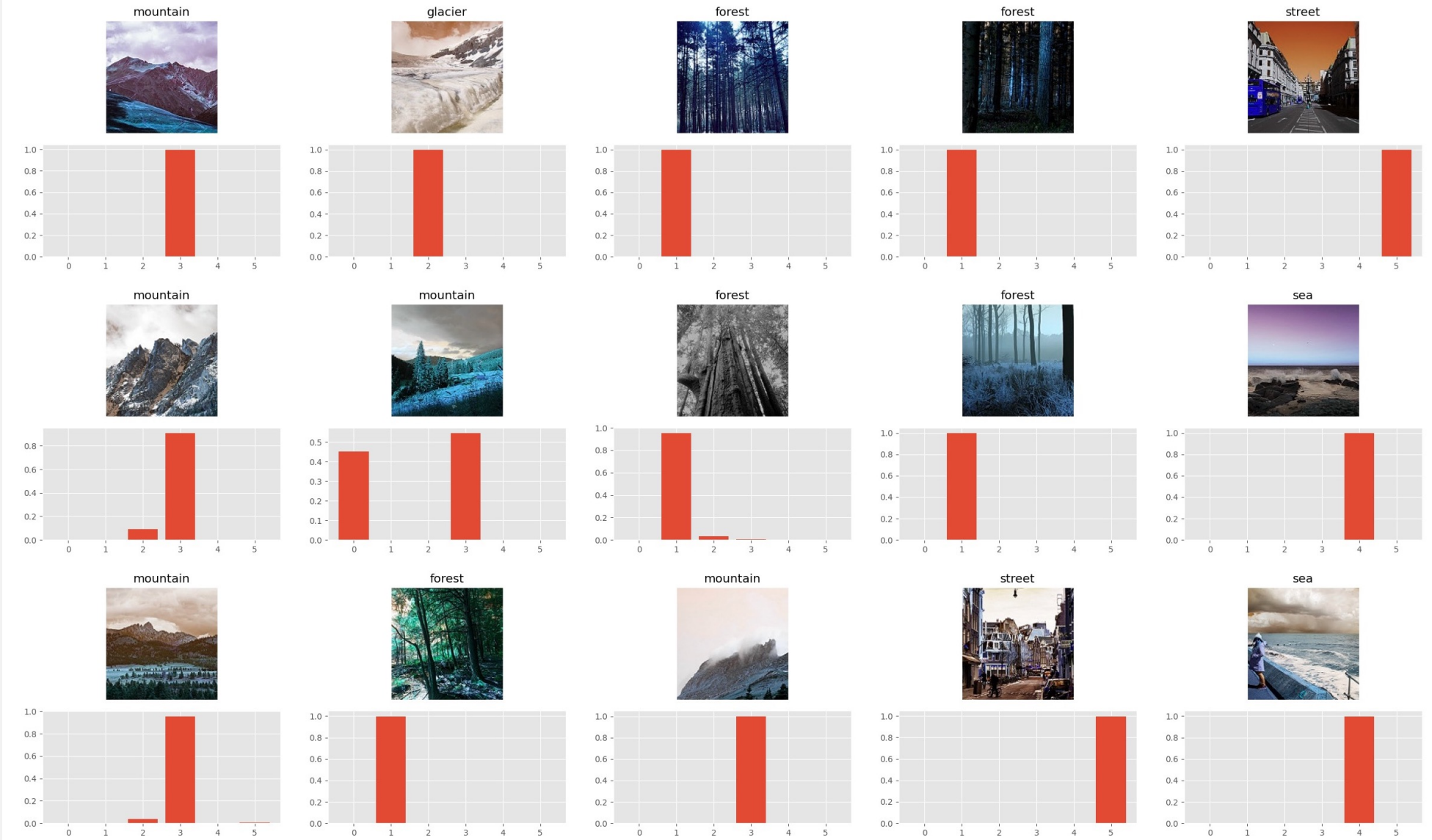
RESULTS_SELF-TRAINED



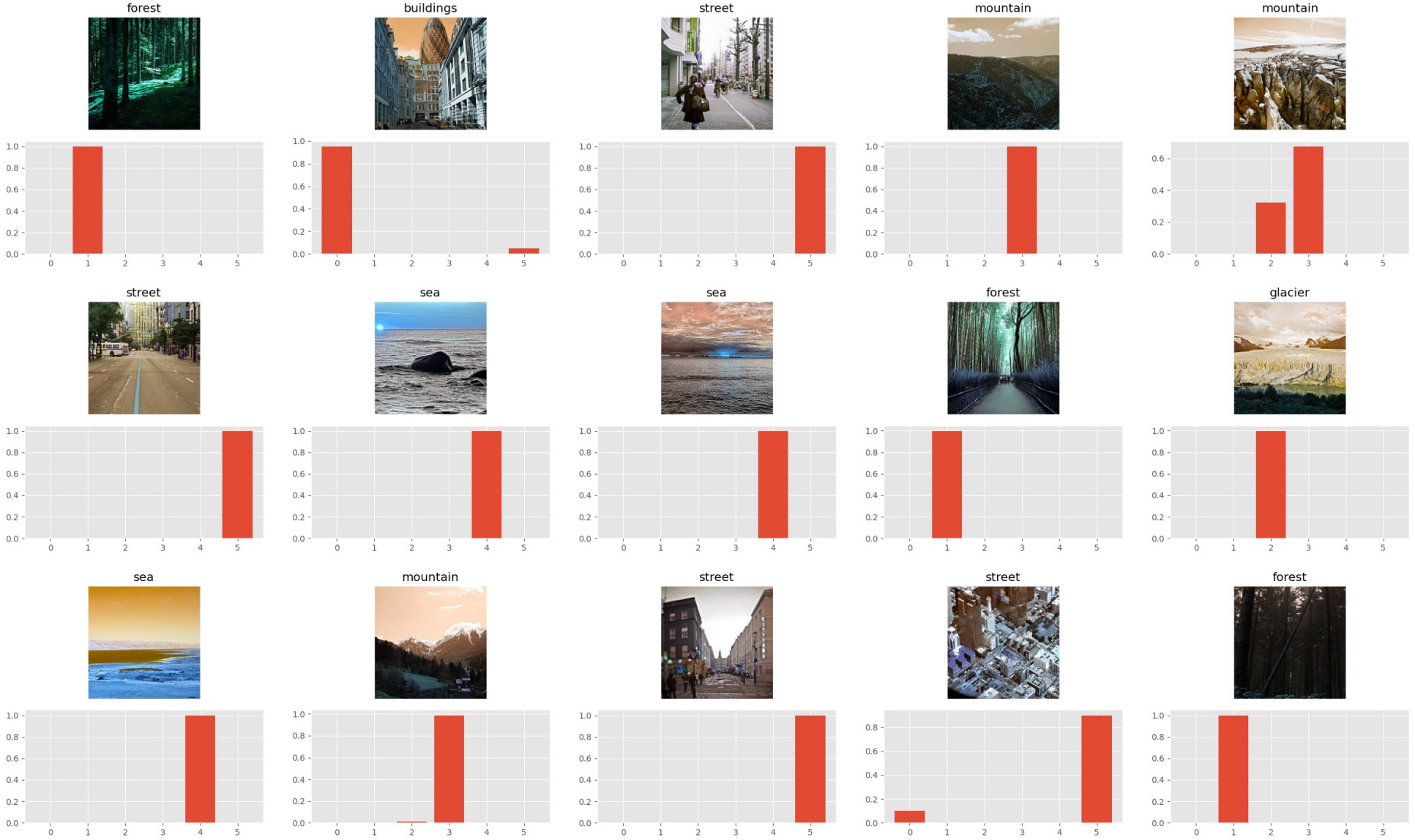
RESULTS_RESNET



RESULTS_VGG16

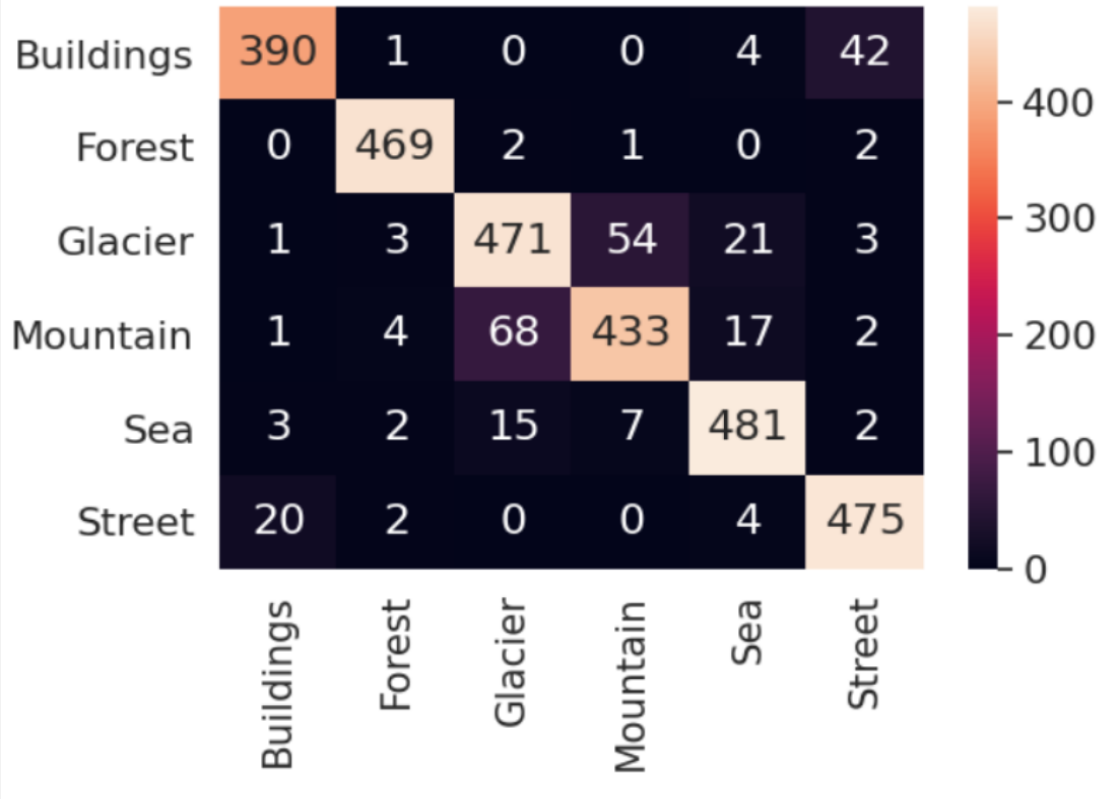


RESULTS_VGG19



ANALYSIS OF RESULT

Confusion Matrix of Best Performing Models :--



So, there seems to be confusion between ‘Glacier & Mountain’ and ‘Buildings & Street’

This was found to be due to nature of training images :--






We tried to solve this problem by taking only those images from the pretrained network whose probability for most probable class label was greater than 90% to retraining the models

But it resulted in loss of about 25% of the training images and could not surpass the previous best model's performance too.

RECOMMENDATION

- Instead of Multi-Class Classification, solving it as Multi-Labelled Classification Problem by adjusting class labels could result in a higher accuracy

THANK YOU!!!