

## I. EVALUATION

### A. Statistics

The annotation table obtained from the image labelling is in the form of rectangle location and dimensions of the bounding box dimensions. The datasets are divided into three sets

- 1) *Training Dataset*: annotation data obtained using the consumer drone and used for training the model.
- 2) *Testing Dataset*: annotation data obtained using the consumer drone and used for testing the model.
- 3) *GeoScience Dataset*: annotation data obtained using the data obtained from the GeoScience Department using a commercial drone.

Bounding box heights, widths and aspect ratios can be obtained from the annotation data. The testing data and GeoScience data will be tested against the training data to see whether they are from the same population. If any populations are the same, the analysis done on one population should have the same evaluation results. If the populations are different, the analysis on both will be different, and the algorithms would need to be adapted to cater for this difference.

1) *Training Dataset*: Training bound boxes have a similar average width and average height pixel sizes. The average size is around 76 pixels, with minimum pixel size of around 28 pixels. The smallest bounding box matches the  $32 \times 32$  VGG16 pixel field of view within the *conv5* output feature map. The highest number of bounding boxes fall under a  $2 \times 2$  VGG16 feature map.

TABLE I  
CONSUMER DRONE BOUNDING BOX STATISTICS - TRAIN DATA

	Mean	Std. Dev.	Min	Max	Population
Width	73.848	28.359	25	206	524
Height	79.293	27.197	31	207	524

Consumer Drone Bounding Box Size - Training Dataset

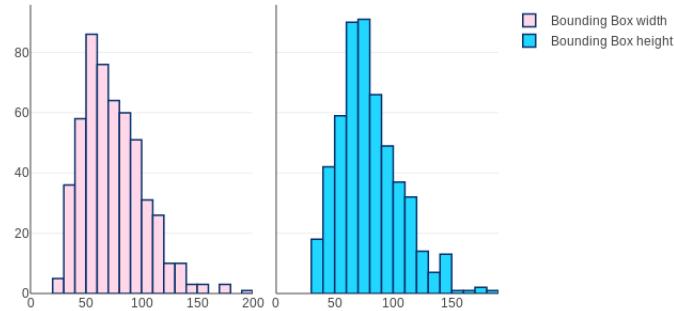


Fig. 1. Consumer Drone Bounding Box Height and Width Histograms- Train data

As per qqplot show in figure I-A1 the data does not follow a normal distribution, so non-parametric tests will be performed to check that samples are obtained from the same population.

The aspect ratio of the training set as shown in the figure I-A1 shows that there is no preference between  $\log(-0.5)$  to  $\log(0.5)$ . This shows that there is no orientation preference of the bounding boxes in the training data set. As can be expected from aerial imagery data.

2) *Testing Dataset*: Testing bound boxes have a minor tendency to be larger in height than in width. The average size is also around 76 pixels, with minimum pixel size of around 29 pixels. Again this is within the field of view of the VGG16 pixel single feature vector of the *conv5* output feature map.

As per qqplot show in figure I-A1 the data does not follow a normal distribution. This confirms the non-parametric tests to be performed on the population sampling.

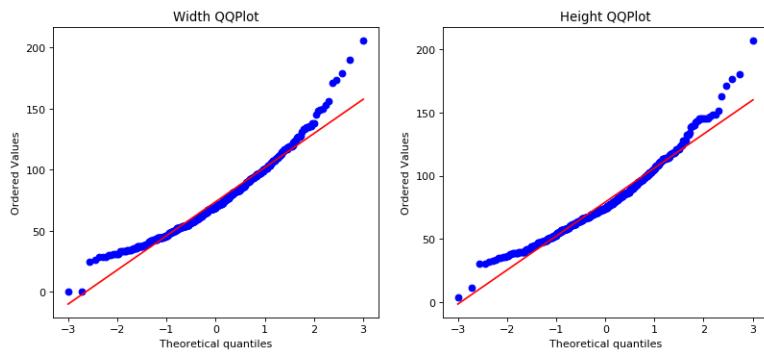


Fig. 2. Consumer Drone Bounding Box Height and Width QQplot - Train data

Consumer Drone Bounding Box Log Aspect Ratio Width/Height (Train)

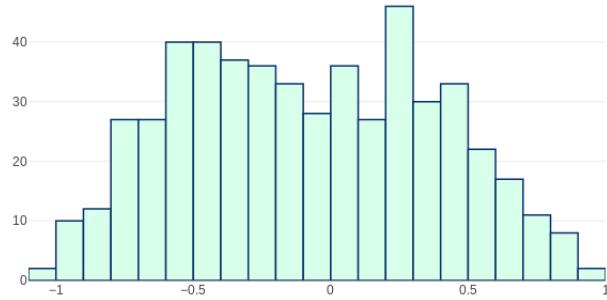


Fig. 3. Consumer Drone Bounding Box Log Aspect Ratio Histogram - Train data

The aspect ratio of the training set as shown in the figure I-A1 shows that there is no preference between  $\log(-0.5)$  to  $\log(0.5)$ . This shows that there is no orientation preference of the bounding boxes in the training data set. The smaller values at the 0 for both the training set and the test set indicate that the objects bounding boxes are rarely a box shape (i.e. equal width and height) which indicates that the objects have an elongated shape. This is to be expected as the images labelled are of beverage bottles and containers, which are predominantly manufactured to be stacked side-by-side, and ergonomically designed to be handled by a single hand.

TABLE II  
CONSUMER DRONE BOUNDING BOX STATISTICS - TEST DATA

	Mean	Std. Dev.	Min	Max	Population
Width	71.654	29.435	26	199	246
Height	81.232	34.203	30	272	246

Consumer Drone Bounding Box Size - Testing Dataset

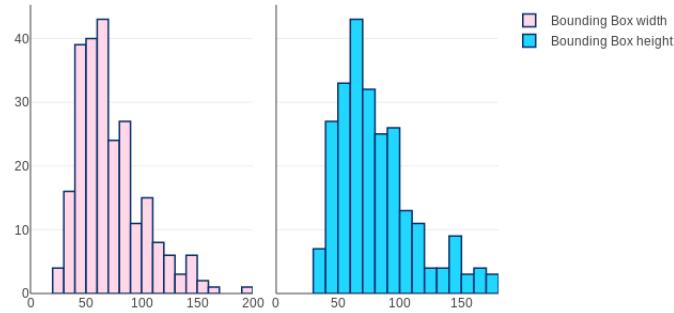


Fig. 4. Consumer Drone Bounding Box Height and Width Histograms- Test data

Consumer Drone Bounding Box Log Aspect Ratio Width/Height (Test)

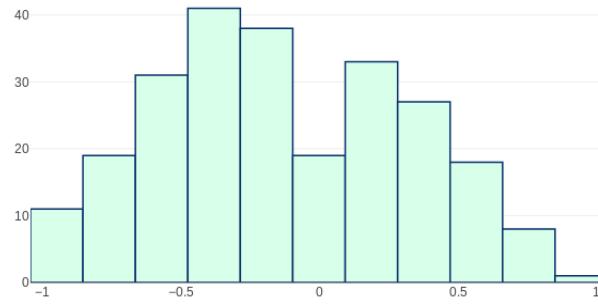


Fig. 6. Consumer Drone Bounding Box Log Aspect Ratio Histogram - Test data

3) *GeoScience Dataset:* The Geoscience annotation also show no orientation preference with equal average width and height. The average size is also around 37 pixels, with minimum pixel size of around 11 pixels. This is a marked distinction from the previous datasets. The smaller bounding box sizes within this dataset might cause difficulties in the VGG16 field of view.

TABLE III  
GEOSCIENCE DRONE BOUNDING BOX STATISTICS - DATA

	Mean	Std. Dev.	Min	Max	Population
Width	36.038	11.921	11	120	793
Height	37.318	10.420	10	82	793

The average size of the bounding boxes of the GeoSciences is half that of the two other datasets.

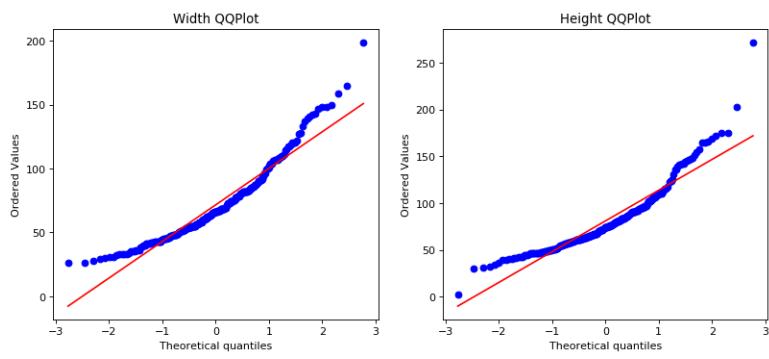


Fig. 5. Consumer Drone Bounding Box Height and Width QQplot - Test data

GeoScience Funded Land Survey Drone Bounding Box Size - Histogram

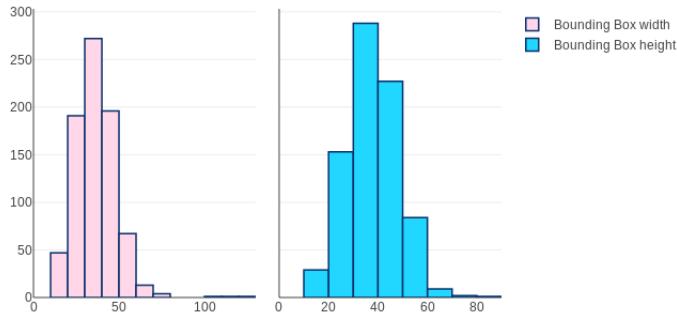


Fig. 7. GeoScience Bounding Box Height and Width Histograms

As per qqplot show in figure I-A3 the data does not follow a normal distribution. This confirms the non-parametric tests to be performed on the population sampling.

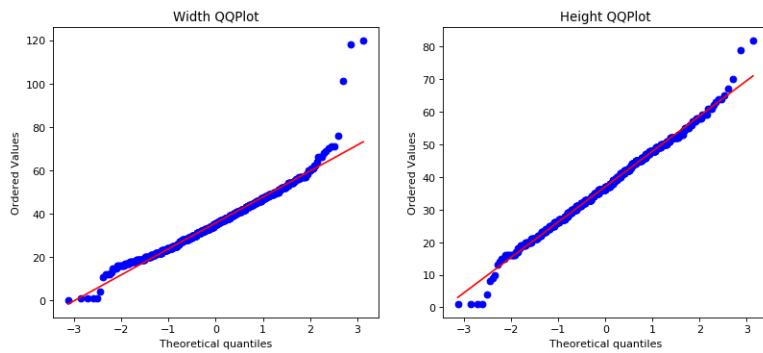


Fig. 8. GeoScience Bounding Box Height and Width QQplot

The aspect ratio figure I-A3 has a similar shape as the train and data set, confirming the nature of the images being labelled.

**4) Population Testing:** The train, test and Geoscience population will be tested for independance. Since the populations are not normally distributed, a non-parametric test will be used. A two-tail Mann Whitney test will be performed. The whole data in the datasets will be used. The scale will be in pixel size. A confidence interval

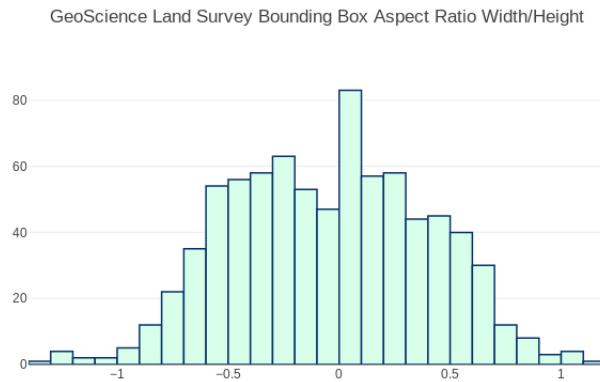


Fig. 9. GeoScience Bounding Box Log Aspect Ratio Histogram

$\alpha = 0.05$  will be used.

The null hypothesis will be formulated :

*Null Hypothesis 1:  $H_0$ :* the distribution of the pixel sizes of the two datasets are equal

*Hypothesis 1:  $H_A$ :* the distribution of the pixel sizes is not equal

The two comparisons will be made between train vs test and train vs geoscience dataset.

5) *Train vs Testing Population:* Testing the two datasets using Mann Whitney results with  $U = 69299.00$  and a p-value of 0.092 (for width) and  $U = 66246.50$  and a p-value of 0.533 (for height) . Therefore we fail to reject the Null Hypothesis  $H_0$ . The train and testing data set come from the same population. This is to be expected as the data comes from the same sensor and using the same land-survey techniques.

6) *Train vs GeoScience Population:* Testing the two datasets using Mann Whitney results with  $U = 383029.00$  and a p-value of 0.0 (for width) and  $U = 400391.50$  and a p-value of 0.0 (for height) . Therefore we reject the Null Hypothesis  $H_0$ . The train and geoscience dataset come from the different populations.

7) *Statistical Results:* The statistical tests show that the geoscience and the consumer drone data have different image sizes. This shows that the surveying techniques and parameters are different between geoscience data and consumer drone data. Comparing the geoscience images to that of the consumer drone, it can be deduced that the altitude of the surveys are different. This can also be the result of the drone operator knowledge of higher camera resolution.

An algorithm which is dependant on the size of the picture for successful recognition, will have difficulty correctly inferring the smaller sized dataset. The ability of the human labeller to correctly label the objects, indicates that the shape information is present. This has to be exploited by the CNN. The algorithm must be scale invariant or specifically tuned to the zoom levels of the geoscience dataset.

### B. CNN Algorithm Testing

Four CNN architectures will be testing for maximum validation accuracy results. The one with highest validation accuracy and highest foreground accuracy, precision and recall will be chosen. Each algorithm to be tested will be preloaded with algorithm weights pretrained on Imagenet. The fully connected layer at the end of the convolutional layers will be stripped off, and replaced by

- 1) A connected layer with softmax activation,
- 2) A connected layer with relu activation and a connected layer with softmax activation,
- 3) Two connected layers with relu activation and a connected layer with softmax activation

The connected layer with softmax activation performs the two binary classifier between foreground and background. The connection layer between the Convolutional Layer and the Fully Connected Layer will be chosen from a Flatten layer, a Max Pooling Layer and a Global Average Pooling Layer. The layer used will depend on

that specified by its relative academic paper when trained on the Imagenet database. If the training procedure does not converge, other layers will be tested.<sup>6</sup>

At this stage the trainable layers will be reserved only for the new fully connected layers. The best algorithm setup will be chosen. Further algorithm tuning will be performed by allowing more layers inside the convolutional blocks to be trained. Incremental training will be performed on previous trained models. Early Stopping will limit the epochs trained.

*1) Dataset Preparation:* One dataset will be compiled for all tests. The dataset will consist of

- 1) Plastic set available from the trashnet database [1],
- 2) Plastic from the training set obtained from the Consumer Drone Survey,
- 3) 10 random background images obtained from the training set obtained from the Consumer Drone Survey,

The data will be fed using Keras ImageGenerator with inbuilt data augmentation algorithms. The options used for this testing are as follows:

- 1) Preprocessing Function: Keras provides algorithm specific preprocessing functions that transform an image to an output tensor of the same shape,
- 2) Shear range 0.2 : Angle of shear in degrees,
- 3) Horizontal Flip : Random horizontal flip of image,
- 4) Vertical Flip: Random vertical flip of image,
- 5) Zoom Range 0.2: Random zoom between 0.8 and 1.2 of the image size.

All algorithms will have a default input tensor size of  $224 \times 224$ . The fill-mode parameter is set to true so that no black areas appear within the image. Later RCNN tuning will test more data augment parameters and parameter ranges.

Validation will only be preprocessed, but no augmentation of this set is performed.

*2) Gradient Descent Optimizers:* RMSProp will be used for all the tests. The learning rate will be tuned according to the training and validation convergence speed. Convergence rate varies between 1e-4 to 1e-6.

### C. Algorithm Test Results

*1) Inceptionv3:*

- 1) : Inception with Max Pooling Layer, FCN 1-layer 1024 and Softmax Classifier

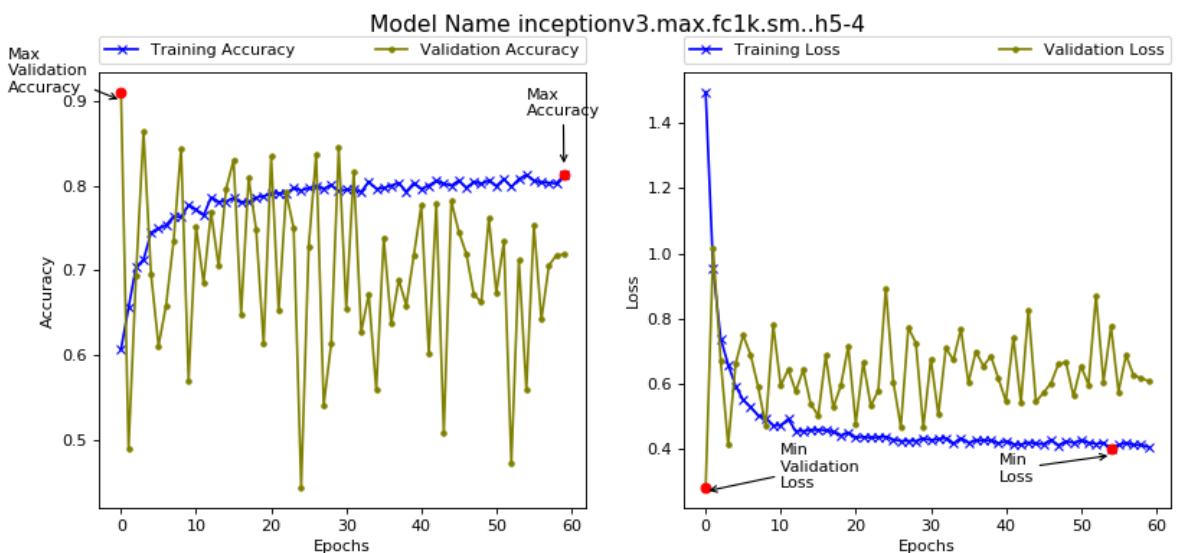


Fig. 10. Inception Training Tasks Max Pooling Layer, FCN 1-layer 1024 and Softmax Classifier

The training procedure shows erratic validation training. Validation Accuracy immediately falls from the first epoch validation accuracy of 0.91. Learning rate is 1e-4, which could be too high for convergence. An average accuracy of value of 0.7 is achieved. The MaxPool layer shows a very erratic validation curve, whilst training curve increases gradually to a limit of 0.8. The MaxPooling layer will be dismissed.

- 2) : Inception with Global AveragePooling Layer and SoftMax Classifier

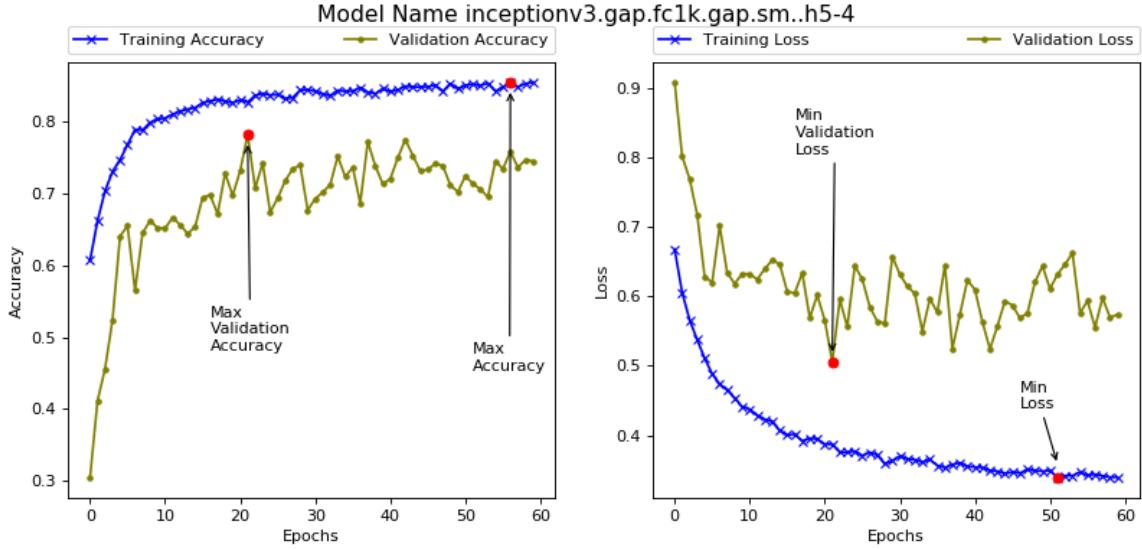


Fig. 11. Inception Training Tasks GAP and Softmax Classifier - training Softmax layer

The InceptionV3 GAP layer maximum validation accuracy achieved when training on 4 layers, is of 0.781 accuracy at epoch 21. The number of trainable layers are increased to 37 layers, whilst using the same weights achieved earlier on.

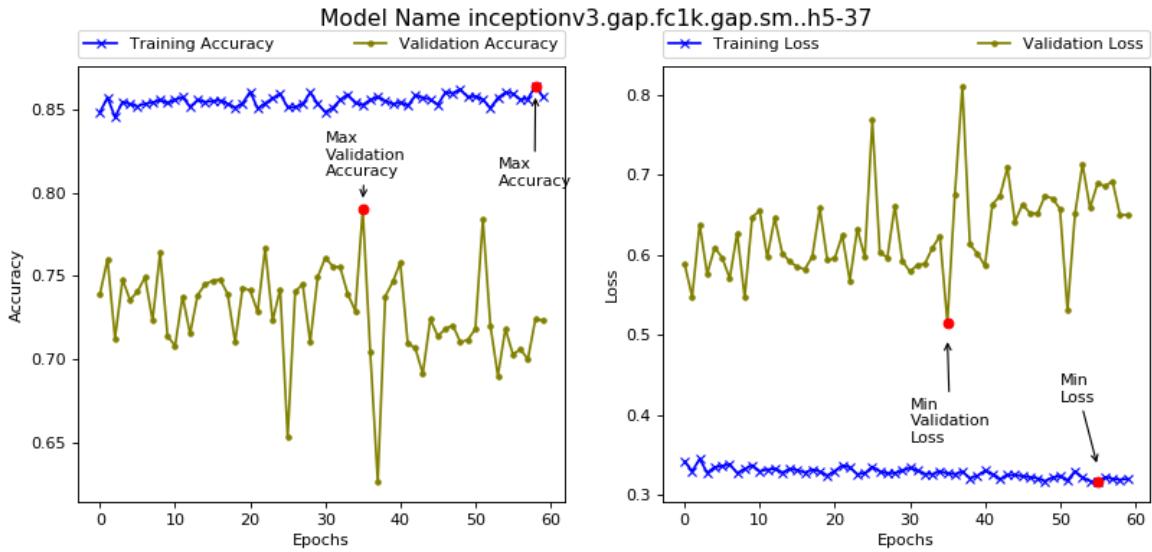


Fig. 12. Inception Training Tasks GAP and Softmax Classifier - training last 37 layers

Validation accuracy increased to 0.79 at epoch 58. The number of trainable layers are increased to 37 layers.

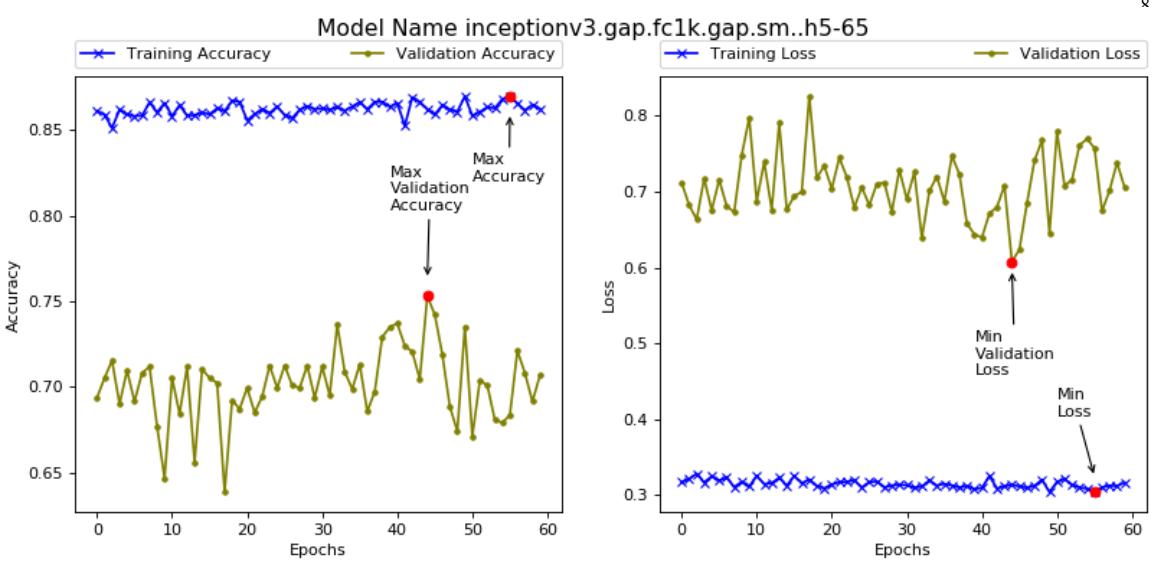


Fig. 13. Inception Training Tasks GAP and Softmax Classifier - training last 65 layers

No further increase in accuracy has been noted, and the validation accuracy levels off at 0.7 mark average. To note the gradual improvement in training accuracy levels, during the training procedure , however the validation is more erratic. This shows a direct neural network to the convolutional output is unable to obtain the classifier features directly. The training accuracy levels off at a 0.87 mark, showing no increase during the last 120 epochs of the total 180 epochs run using a Global Average Pooling Layer.

Incrementing the FCN to two layers produced an neural network without high value validation accuracy convergence.

## 2) Resnet-50:

3) *MobileNetv3*: A successful mobilenetv3 implementation will help us increase the speed of all subsequent inferencing. The default depth multiplier of 1 and default number of filters in the layer will be used as per paper. Detailed mobilenetv3 results are available in Appendix A.

Mobilenetv3 test first using the Convolution Block and Softmax classifier directly to the two classes denoting foreground and background. The conv blocks and the classifier will be linked using Flatten or GlobalAveragePooling layer. Learning Rate 1e-5 used with RMSProp optimizer. Training Batch size 150 and Validation batch size 50. EarlyStopping is used as a regularizer with patience of 10 epoch training iterations. Results found in table IV.

TABLE IV  
MOBILENETV3 TRAINING SCHEDULE

MobileNetv3 architecture	Maximum Accuracy	Epoch
Mobilenetv3+Flatten+SM	0.8310	6
Mobilenetv3+GAP+SM	0.8233	55
Mobilenetv3+GAP+FC1K+SM	0.8442	12
Mobilenetv3+GAP+FC2K+SM	0.8674	10
Mobilenetv3+GAP+FC4K+SM	0.8645	15
Mobilenetv3+GAP+FC2K+FC2K+SM	0.8188	50
Mobilenetv3+GAP+FC2K+SM (conv13)	0.9514	50
Mobilenetv3+GAP+FC2K+SM (conv12,conv13)	0.9514	50

The flatten layer in the first training gave an erratic training curve, so a GlobalAveragePooling layer was used for the rest of the schedule. A fully connected layer was introduced with varying widths (FC1K = 1024 neurons, FC2K = 2048 neurons and FC4K = 4096), followed by dropout layer at a rate of 0.5 and ending with a two-class softmax classifier(SM). The best accuracy achieved was for the Mobilenetv3+GAP+FC2K+SM. An extra hidden layer of 2048 did not improve accuracies. The only trainable layers were the Fully Connected Layers

introduced, so the schedule started training the deepest convolutional block layers of Mobilenetv3. Training *conv13* and *conv12+conv13* resulted in the same maximum accuracy so no further testing were performed on the dataset.

#### 4) VGG-16:

#### D. R-CNN Evaluation

1) *Test 1: Sliding Window - Baseline:* Sliding Window uses the convolutional neural network over the whole image. Tiles of 224x224 size are cropped from original image from a 192 grid ( $192 = 224 - 32$ ), with 32 being the overlap allowed between the two tiles to allow for detection of objects at the edge of the tile. The last row and last column are recalculated to allow for mismatching image height and width to grid.

The tile passed to the VGG16 inferencing engine, which outputs a 2 unit tensor having values for [nolitter,litter]. The value of litter is stored in the detection grid. Bounding Box co-ordinates are obtained using tile co-ordinates for the tiles which inference results is higher than a preset confidence level. The bounding box predictions evaluated against the ground-truth values.

VGG16 Training: A new VGG16 architecture will be trained on the dataset :dataset.v1. *Conv5*, *conv4* and *conv4* will all have trainable layers. Two fully connected layers with relu activation and a softmax layer (2 classes) will be appended to the VGG16 convolutional layer output. The CNN will be trained using checkpoint per epoch, early stopping based on Validation Accuracy metric with waiting for 10 epochs for validation accuracy improvement.

TABLE V  
DATASET.V1

	<b>Litter</b>	<b>Background</b>
Training	5174	2672
Validation	208	1172

Litter has been obtained from annotation data, whilst terrain obtained by sampling 10 images per scene, and vetted for presence of plastic or glass containers. Litter subdivided into 70% training and 30% validation. However, 70% training data has been augmented manually to 9 shifted positions.

VGG16 architecture training: The validation accuracy obtained was of **0.9833**.

TABLE VI  
VALIDATION CLASS REPORT

	<b>Litter</b>	<b>Background</b>
Precision	0.97	0.99
Recall	0.95	0.99

The precision/recall values are quite high for all the dataset. This maybe due to the limited number of cases present in the dataset. One has to note however that in RCNN the number of times the inference is done, where terrain is to be expected has a probability of 2000-3000 more than litter instances. Therefore the precision/recall obtained in the terrain 0.99/0.99 will have much more impact than the lower but less likely litter detection precision/recall of 0.97/0.95.

In a 4000 tile inference, 1% of incorrect predictions would provide 40 False Positive predictions. In a typical scene 3 litter objects, would give us an R-CNN precision value of  $\frac{3}{3+40} = 0.069$

Consumer Drone Testing Results: **Average Precision 0.14** at IoU of 0.1. The Recall rate is 0.922 (objects located within a bounding box) or 0.859 (bounding boxes predicting objects). Due to the large size of the bounding box with respect to the size of the detected object there are occasions of a bounding box covering two objects. Two recall rates are being reported to denote the discrepancy between the two possible calculations.

This is a high recall rate indicating that litter is being detected successfully, and is very close to the validation litter rate detection. However the precision rate is very small at **0.0586**, as per example above.

1) Correct Predictions

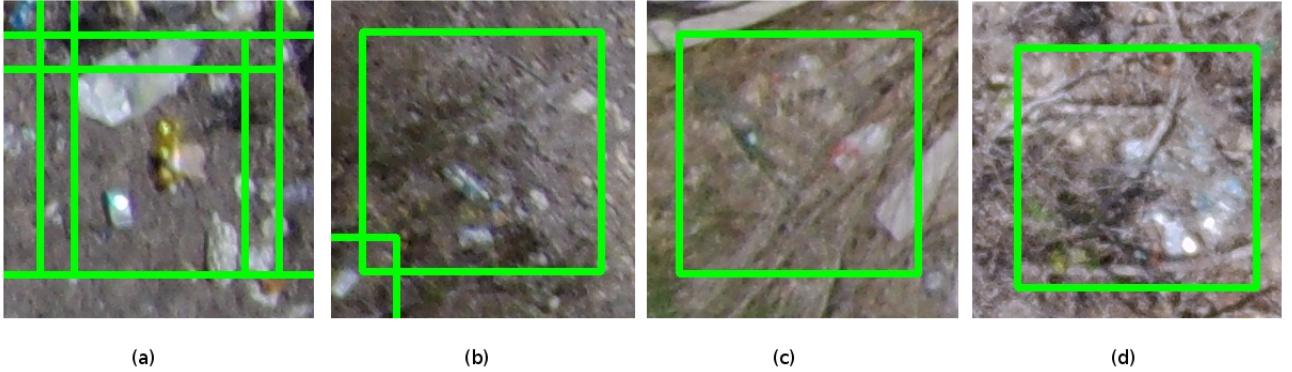


Fig. 14. Sliding Window Correct Predictions

Figure 14 shows successful predictions on the Consumer drone test set. The large sliding window size and stride do not hinder the correct inferencing. The ratio of the size of the detected object is comparable to the inferencing window. Figure 14c and 14d show occlusion and deformations being detected adequately. To denote the specular aspect of the objects being detected.

2) Incorrect Predictions But Plausible

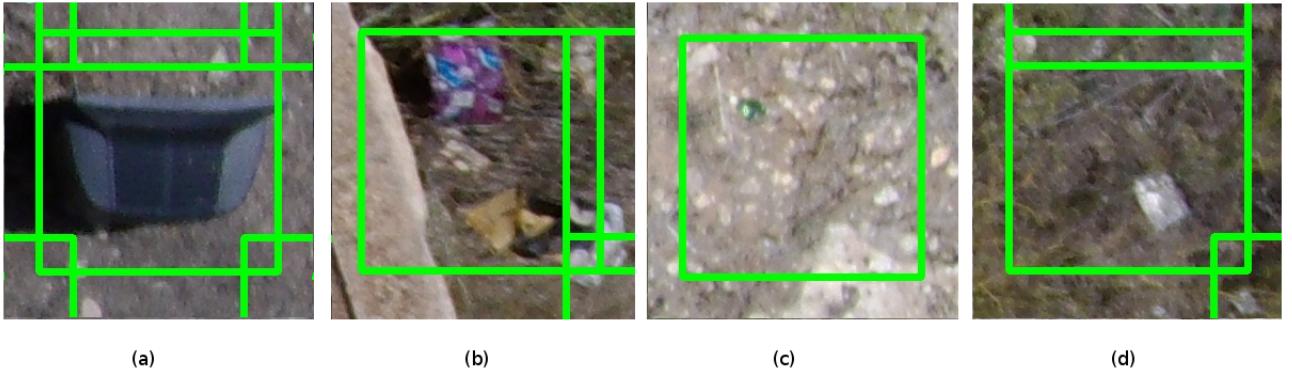


Fig. 15. Sliding Window Incorrect but Plausible Predictions

Figure 15 shows a number of predictions that have not been annotated as beverage bottles, but which the algorithm has marked them as litter. These objects are frequent in the detected dataset. These artefacts found in outdoor imagery degrade the precision value, even though they could be successfully tagged as litter, though not beverage bottles or containers. The common aspect of these artefacts is that they are man-made objects, distinct from the background. Objects small as in figure 15c are very frequent. However the ability to detect these small objects, although enhancing the shift invariance of the algorithm, can have a major detrimental effect on the precision value, as anything with this feature can be tagged incorrectly as litter. Such reflective objects could be stones, sea reflections, glass fragments, etc....Human annotations of these small objects is very difficult.

3) Incorrect Predictions that can be ammended

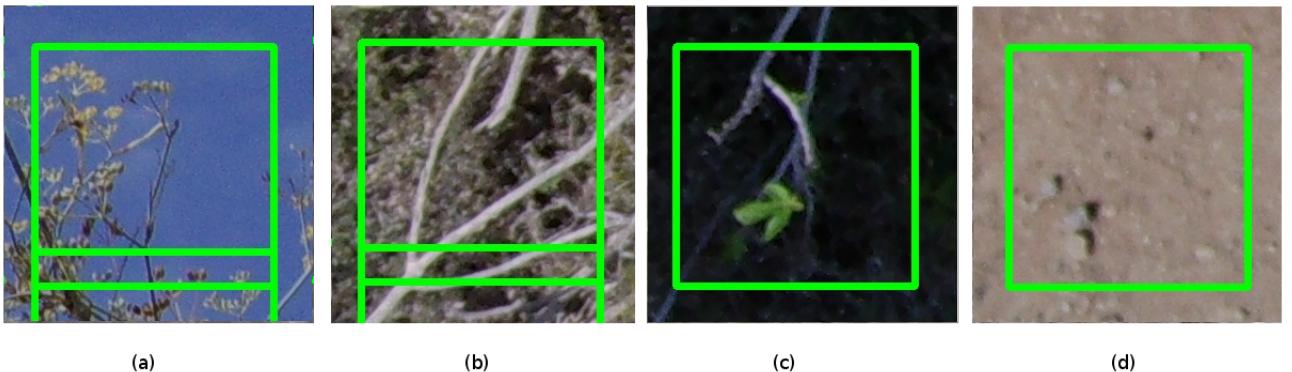


Fig. 16. Sliding Window Incorrect Predictions that could possibly be corrected

Figure 16 shows examples of uncorrect predictions that could possibly be corrected. Figure 16a shows a sky scene with branches that have been mistaken by the algorithm. This picture is evidently not an aerial image because sky can never be shown from a UAV with its camera shooting downwards. Accidental images such as these (during takeoff or landing of the drone) are possible and could be mistakenly placed within the dataset. The imagery can easily be removed by filtering *sky* image tiles within the pictures. Figure 16 (b) and (c) are scenes depicting vegetation and branches that also have been incorrectly misdetected. The regular features of the twigs could be affecting the CNN. However, different to the *sky* images, plastic and litter can be found within *vegetation* imagery, therefore handling vegetation must be done by the RCNN. Figure 16d is an image with no clear plastic or man-made features. There is no clear explanation why the RCNN mistook this for litter. However, like *vegetation* the RCNN could be trained to detect this type of imagery as terrain.

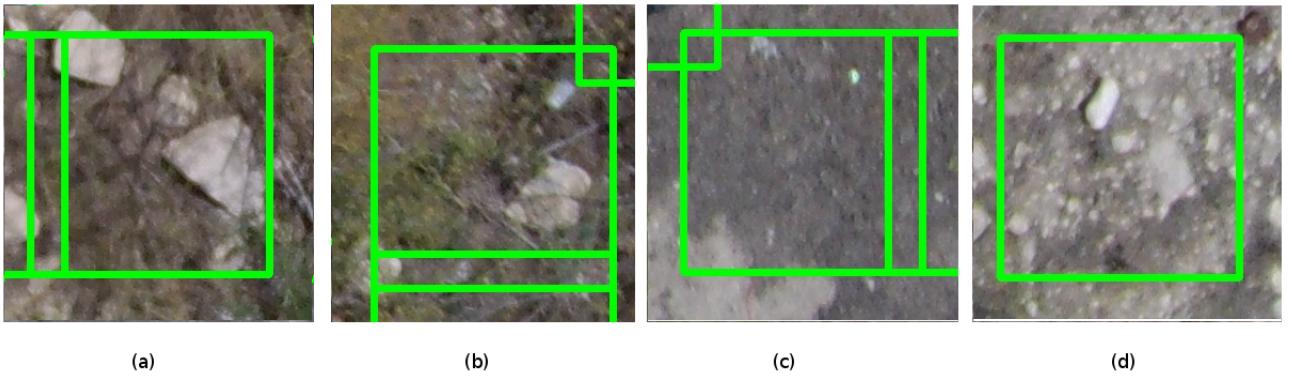


Fig. 17. Sliding Window Incorrect Predictions

Figure 17(a),(b) and (d) are images of stone incorrectly labelled as litter. Figure 17(d) particularly depicts stones with a specular features and shapes very similar to plastic or glass bottles. Surprisingly some scenes contain a lot of these stones, to the detriment of the algorithm precision. Figure 17(c) shows a very small feature, as described in figure 15c affecting the classification of the CNN. This is caused by the high sensitivity to litter that has been designed within the dataset. Such example caused by high litter sensitivity will cause a high False Positive number. Unfortunately regular features can be detected anywhere within the outdoor imagery but not be result of manufacture but by stone formations or shadows.

**GeoScience Drone Testing Results:** **Average Precision 0.02** at IoU of 0.1. The recall rate is 0.896 (objects located within bounding box) or 0.8225 (bounding boxes predicting objects). The average precision is much lower than the consumer drone test results. The recall rate is lower but one can still say that out of the annotated

number of bottles a good percentage is being detected. This indicates that the algorithm has a good degree of scale invariancy, since the statistics have shown that the size of the bounding boxes are different. The precision of the R-CNN is lowering the average precision value. At a precision rate of 0.0091 one can say that only 1 in 100 predictions is correct. This partly stems from the larger number of tiles to be predicted, and partly due to the new terrain which is causing the algorithm to wrongly detect litter.

### 1) Correct Predictions

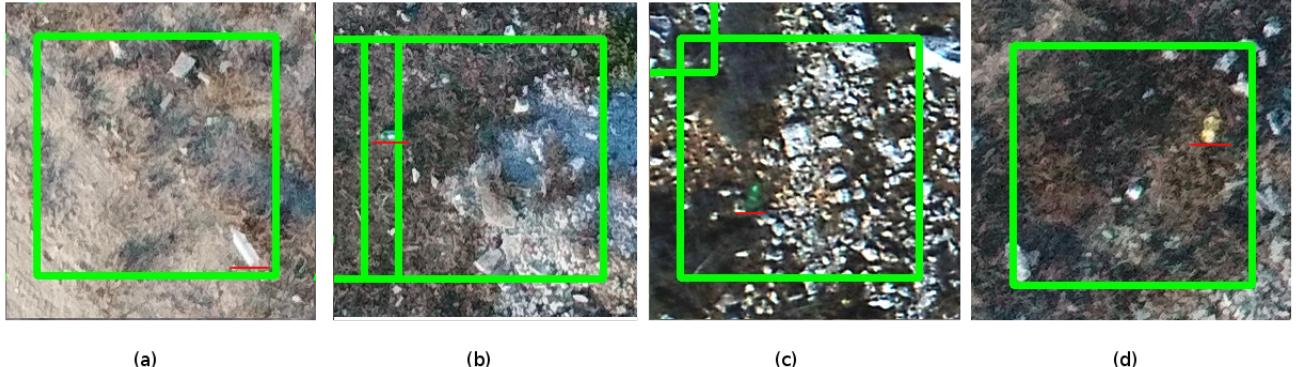


Fig. 18. Sliding Window Correct Predictions from GeoScience Survey

The red underline is to highlight the position of the object

Figure 18 shows the correct predictions by the algorithm that have been detected. As expected the size of the objects are much smaller from the objects found in the Consumer database set. To note that the image quality is superior than the drone imagery, and the definition of both the objects and the terrain is better. The quality difference might have an impact because of the regular features learnt by the CNN algorithm. The CNN is invariant of the position of the object, and the overlap in figure 18(b) shows two nearby tiles detecting the same object.

### 2) Incorrect Predictions but plausible

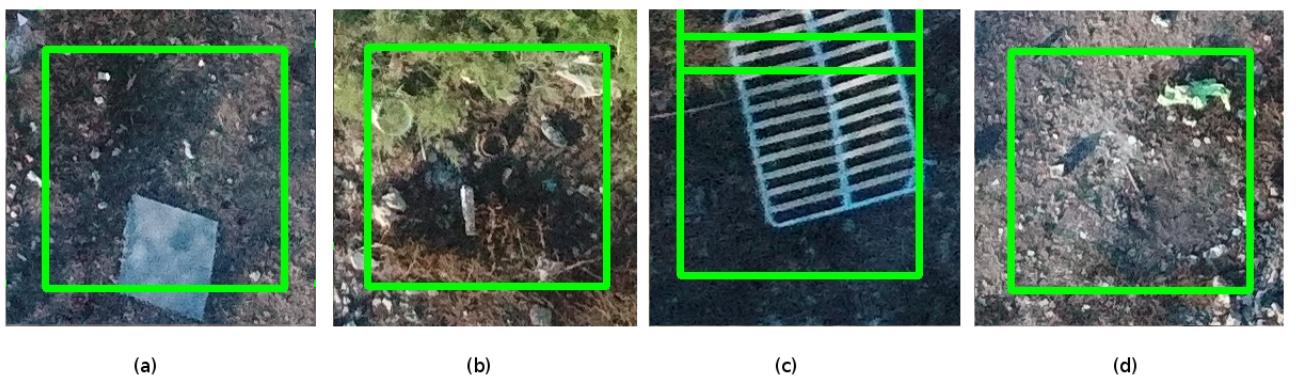


Fig. 19. Sliding Window Incorrect but plausible predictions from GeoScience Survey

Here again we see that the algorithm is detecting very linear features and man-made shapes as being litter. Although one can correctly define them as litter the CNN algorithm was trained on beverage bottles and containers. The smooth and linear edges found within these objects are being extrapolated by the CNN as bottles. The CNN, as in case of 19(b) has managed to detect occluded objects.

3) Incorrect Predictions that can be ammended

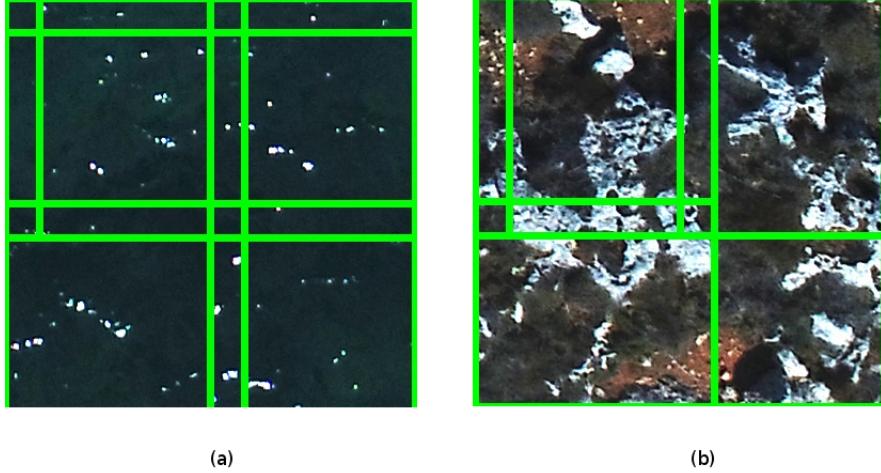


Fig. 20. Sliding Window Incorrect predictions from GeoScience Survey (a) sea cover (b) rocky, grass and soil terrain

The geosciences dataset has a number of terrains not present in the consumer drone dataset. This has affected majorly the CNN inferencing ability on this dataset, why the precision rate for the GeoScience dataset is so low. The terrains in figure 20 are an example of this imagery that are shown to be detected litter. Figure 20a is sea and the reflections on the ocean are being mistaken as specular reflections of bottles and glass. A large amount of incorrect predictions are being made, since the around 20% of the survey is carried out over water. Since the application is for land-detection of litter, we can easily omit these by filtering these images out. With regards to 20b the CNN also is having a very large number of misdetections caused by unknown terrain. Removing such a terrain should be performed by modifying the litter detecting CNN, since plastic within this terrain should also be detected, and cannot be filtered out.

#### E. Test 2: Class Activation Mapping

A human annotator will deduce the size of the objects once he starts reviewing the images from the aerial shots. The RCNN as shown is shift invariant to the cater very small objects as shown in figure 15c and figure 17c. However, through Class Activation Mapping one can determine the object size from CNN inference. A preliminary test on CAM methods was performed to compare it with the sliding window technique. The CAM output will be later on used to try to increase the precision values of the aerial-litter CNN. The CAM method allows the algorithm to perform a weak object localisation based on the activation output. This allows use to reduce the Sliding Window predicted box down to smaller and more accurate proposals.

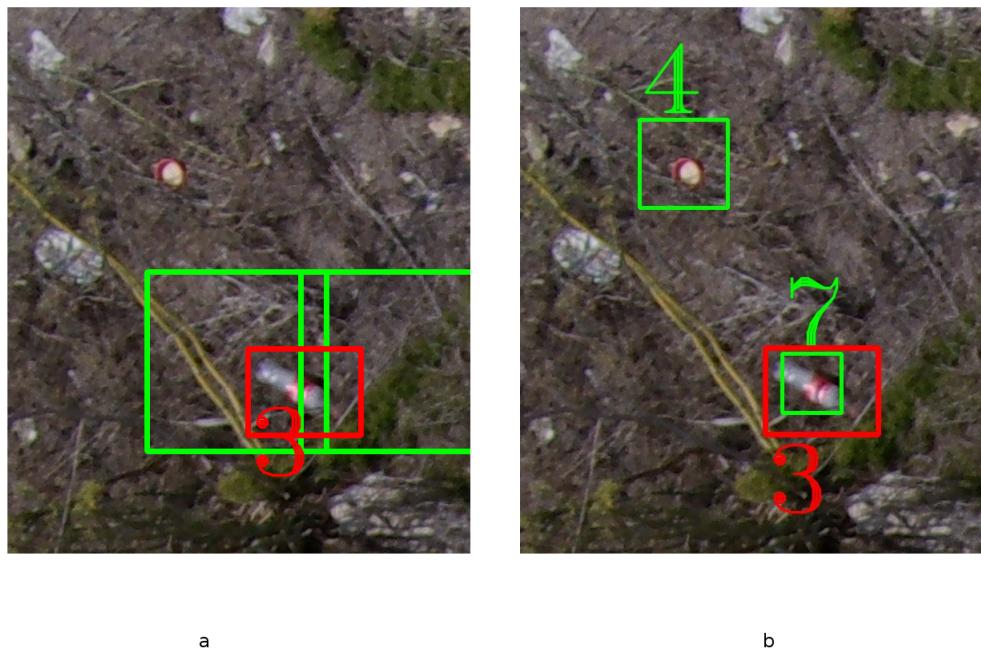


Fig. 21. Prediction output comparison (a) VGG16 RCNN using sliding window (b)VGG16 using CAM prediction boxes

Note that the red bounding boxes indicate the ground-truth, whereas the green bounding boxes are the bounding boxes predicted by the algorithms. Note in figure 21 that the CAM approach produces prediction boxes more accurate than the human annotation. This will allow us to increase the IoU threshold from a small image setting of 0.1 to a normal setting of 0.5, without degradation in performance. The CAM approach was introduced to decrease the operational times for inferencing. The CAM stride was double that of the sliding window, downscaling the image by a factor of two. In spite of the downscaling, the proposals obtain still more accurate and close to the ground-truth than the sliding window technique.

The CAM output is followed by the VGG16 inferencing engine. This was done to validate the weaker classifier the VGG16-CAM. The two filters reduce the recall rate, as both drop their False Negatives. Table VII shows an increase in precision from 0.0584 to 0.254, however a reduction in recall rate.

TABLE VII  
VGG16-CAM RESULTS

Dataset	Average Precision	Precision	Recall
test	0.18	0.254	0.647

A number of tests will be performed on the sliding window, in order to increase the precision. The CAM method will be used in later algorithms to create an more precise locator.

#### F. Test 3: Dataset Engineering

Inspecting the bounding box predictions of the sliding window there are a number of terrains that are being erroneously tagged as plastic as previously described.

As Chawla et al [2] the sensitivity of an algorithm to the detected object can be tuned by altering the amount of class examples during the training of a machine algorithm, in our case the VGG16 inferencing engine. We will

treat the number of examples in the dataset as being a hyperparameter that we can tune for optimum algorithm performance. The softmax classifier is being trained to discriminate litter beverage bottles versus the background terrain. The VGG16 is trained on images containing the object against images containing no objects. The hindsight beneath dataset engineering is to introduce more terrain examples of what the algorithm is continuously tagging as plastic.<sup>15</sup>

The amount of plastic examples that are available to the dataset is fixed (unless we carry out more surveys). Therefore, as more terrain examples are included in the dataset, the ratio for the number of examples plastic vs non-plastic will verge to the non-plastic (or terrain). We will expect that as less terrain gets erroneously classified as plastic, the number of true positives will decrease accordingly. So by dataset engineering we will be expecting smaller recall rates however at a greater precision value.

The terrains being identified are shown in table VIII

TABLE VIII  
ADDITIONAL TERRAINS EXTRACTED FROM CONSUMER DRONE SURVEYS

Terrain	No of Samples	Comments
Sea	5902	Filter
Sky	1214	Filter
Vehicle	65	Filter
Stone	1255	Dataset
Vegetation	1020	Dataset

Each terrain will be included in the original dataset or used in the pre-filter. The original dataset is shown in table IX. To note that currently the training dataset shows a slight imbalance to the plastic dataset. This will produce a classifier which is more likely to tag an image as a plastic object i.e. a classifier having a high-rate of False Positives. As more terrain is introduced this is imbalance is reduced, and in later tests shifted towards a high terrain content.

TABLE IX  
DATASET V1 NUMBER OF SAMPLES PER SET

	Training	Validation
Plastic	5174	208
Terrain	2672	1172

*1) Test 4.1: Pre-Filter:* Initially sky, sea and vehicles were to be included in the dataset as described above, for hard-classification. However, to note that these terrains naturally occlude any presence of plastic. The project assignment was to locate litter found on land. Therefore it was decided that terrains to be removed completely from the search. This resulted in a number of benefits namely:

- 1) Performance: The filtering can be performed once to produce new images with sea and vehicles occluded. All other tests would be performed on the new filtered images. By detecting the masking out of an image, later classifiers can be made to ignore the patch and reduce the penalty of one classification per filtered tile.
- 2) Accuracy: A binary classifier would be able to obtain better discriminating image features to filtering sky/sea/vehicle versus all other terrains, rather than a classifier trying to filter plastic versus terrain (where terrain contains road, garigue, coastline, trees, shrubs, stone, construction, fields, etc....).

# TODO

Fig. 22. Sea and Vehicle being mistaken as Beverage Bottles

The filter was performed using the same VGG16 convolutional block with two fully connected layer and a softmax classifier. A new image would be created using the older image with tiles blacked out, wherever a positive detection of sea terrain or vehicles would result. Future inferencing algorithms would detect beforehand a black tiles, and would skip that tile prediction.

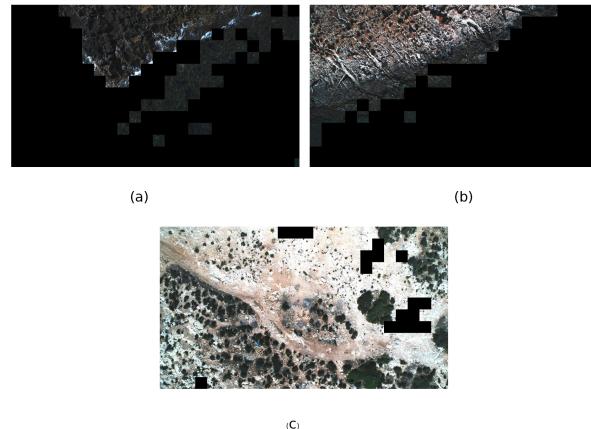


Fig. 23. Output of PreFilter on scenes from dataset of GeoScience Survey

The output of the prefilter on the GeoScience survey images shows that the sea surface has been removed, though not completely. In figure 23a the low level sea one can still see the sea bed, and therefore the CNN algorithm determined that this was not the sea surface. To note that the images used for pre-filter were also obtained using one survey of the Consumer drone. Unfortunately, this survey did not contain low level seas, therefore there were no examples such as this sea-type. Figure 23c shows that the filter has also misclassified some terrains as being sea. Following this misclassification of terrain the prefilter could possibly remove the chance of detecting plastic litter. A test confirmed that no tile that was filtered out contained a ground-truth bounding-box.



Fig. 24. Output of PreFilter on scenes from dataset of Consumer DroneSurvey

Figure 24 shows the correct removal of sky and vehicles from the Consumer drone surveys. Sky images can be erroneously inserted within surveys. This is possible because on initiating the survey (or drone flight) or during camera movement, some sky images might be erroneously inserted in the data. Figure 24a shows two vehicles removed from the scene.<sup>17</sup>

2) *Test 4.2: Dataset Engineering:* The pre-filter is useful only in images containing terrains that we do not expect or not require to detect litter. Finding beverage bottles in other locations is a possibility, therefore filtering out parts of images is unsuitable. The idea of dataset engineering is to ensure that the hard-classification of the CNN algorithm is more susceptible to features found in beverage bottles rather than features that could be common to terrain and bottles. This fine-tuning will be done by increasing the number of samples that the CNN is tagging as False Positives. The CNN from the first set is identifying long shadows from stones, regular features from walls and stone slabs, and branches from trees. Regular features like pipework is also being mistaken as

## TODO

Fig. 25. (pipe/stone/walls/trees) Terrain being mistaken as Beverage Bottles

The amount of samples inserted into the dataset *terrain* is given by table VIII.

- 1) stone
- 2) vegetation
- 3) all

Following the results from the dataset engineering, the following tests will be performed with the dataset using all extra terrains included. The aim is to always look for the best method which increases the precision value whilst maintaining recall value.

### G. Test 5: Field of View

The training done on the land surveys obtained from the consumer drone has obtained results that have not been replicated on the GeoScience dataset survey. From the statistics we have seen that the objects found in the latter surveys are half the zoom-scale found in the consumer drone bounding boxes. This difference might be the reason behind the difference in results. To check whether the algorithm is not shift invariant enough for the smaller-object dataset, we perform two modifications to the CNN algorithm to improve its shift invariancy.

- 1) Upscaling the images found in the geoscience dataset: Since the GeoScience bounding box mean is half that of the Consumer Drone bounding box mean, the new algorithm rescales the tile images with a zoom factor of 2. The sliding window algorithm has been modified so that each tile measures  $112 \times 112$  and is resized using OpenCV using a bilinear interpolation. The pixel height to width ratio is maintained. The bilinear option was chosen as this was the fastest rescaling option which maintained smoother edges, i.e. it reduced the presence of high frequency features obtained by jagged edges.
- 2) Increasing the training zoom parameters in the Data Augmentation techniques. Another option available is to increase the shift invariancy of the model directly through data augmentation. The *ImageGenerator* by Keras provides automatic scaling of the samples given to the algorithm. In the first training of the sliding window, by setting the *ImageGenerator* *zoom\_range* to 0.2 i.e. images could be scaled from 120% of the image size down to 80% of the image size. Knowing the difference of bounding box images size, the *zoom\_range* factor was increased to 0.5 i.e. images to be scaled from 150% down to 50% of the original image size. It is

expected that this type of shift-invariance would be less effective than the upscaling of the images because only part of the sample would be scaled to the desired zoom factor.<sup>18</sup>

#### *H. Test 6: Further Data Augmentation Tuning*

The initial dataset data augmentation, some parameters were set to get a minimal amount of data-augmentation. The parameters are shown in table X

TABLE X  
DEFAULT DATA AUGMENTATION PARAMETERS USED

Augmentation Technique	Parameter setting
Shear Range	0.2
Zoom Range	0.2
Horizontal Flip	True
Vertical Flip	True

Additionally the plastic dataset was augmented by shifting the bounding box onto nine shift positions. Otherwise the neural network would have expected plastic to be located only in the middle **ninth** box. The advantage of performing shifts outside the data augmentation provided by the ImageGenerator, is that more real-image terrain is included in the end image. Data augmentation would require to fillup the empty space resulting from the shift by using the same data available within the image. As will be shown later on, real-world data is more useful for the CNN training than synthetic data.

Tests where performed to see whether any improvements can be made by increasing the parameter range of the available augmentation techniques. The end evaluation is always the average precision, precision and recall values obtained by the RCNN.

- 1) Height and Width Shift: The nine sample shift done within the dataset still has definite positions of plastic locations. We do not have this assurance that plastic will be available solely in these nine positions. Therefore to perform more fine tune shifts, we use the horizontal and vertical shifts available within the ImageGenerator. The height shift and width shift parameters are set to 0.25 i.e. 56 pixels.
- 2) Shear Range: The shear angle in degrees performs vertical shifts, useful when seeing an object from a different vertical angle. Shearing does reduce image quality and excessive ranges might reduce the shape data extensively.
- 3) Rotation Range: Initially we opted only to use flipping and mirroring for data augmentation. These data augmentation techniques have the advantage of maintaining image quality and do not introduce terrain alterations (by filling in empty spaces left behind from non-90°rotations). A possibility that the small number of examples available in the dataset might be reducing the detection of images resulting in False Negative. To this end the rotation range will be tested for 22° and 45°. Since horizontal and vertical flipping will still be performed, these degree ranges will be sufficient to cover full rotation of the image samples. The Keras implementation of the ImageGenerator the parameter is the range of random rotations performed on the images.

#### *I. Test 7: Normalisation*

In machine learning it is common practice to normalise the data before training the machine algorithm. In image processing this is performed per channel of RGB input. The ImageGenerator implementation in Keras provides two normalisation methods featurewise and samplewise centering. In samplewise centering the normalisation takes the mean of the sample per channel and subtracts it from each image used during training. In featurewise centering the mean of the whole dataset is used to make the normalisation. Like Z-scale normalisation, these techniques also divide the result by the standard deviation of the sample. These parameters are booleans sent to the ImageGenerator function. To note that featurewise normalisation was reported to have unsolved issue within its implementation, so the results might be inconclusive.

#### J. Test 8: Reduce Litter Sensitivity

Up until these tests the only two processes that have been successful in improving the Average Precision results were the pre-filtering and the increase in data available inside the terrain class. To test the criteria of whether it was correct to choose only 10 samples per scene, the sliding window CNN will be trained on increasing number of terrain samples, added to the existing dataset. Random tiles will be chosen from each each scene. The tile will be tested not to coincide with existing ground-truth bounding boxes and between proposed terrain tiles themselves. The tiles will be vetted again to remove any possibility of plastic litter (that would have been possibly missed or litter not falling under beverage bottle class). The CNN will be trained using the same plastic samples.

To note: the CNN will be trained and early stopping used based on validation accuracy. As more terrain samples are included in the dataset, the plastic class will move from a majority class to a minority class until with 100 terrain samples, the plastic becomes a very small percentage of the whole training. Also the validation set, ever greater emphasis will be made on correctly detecting terrain rather than litter. The end evaluation will be done by the overall R-CNN average precision, precision and recall values.

#### K. Best R-CNN - GAP size

#### L. Using GeoScience as the training set

#### M. Comparison with Peer Small Object Detection

*Indoor Objects:* Krishna et al [3] in their improving of small object detection data, improve on R-CNN by including an upsampling autoencoder with skip connections to upscale low-resolution data. The inferencing is done by a Faster RCNN with VGG16 algorithm. The dataset is based on internal scenes using internal objects, with a lesser degree of variability than outdoor terrain, and the CNN are trained to categorise the objects between themselves rather than a foreground/background technique used by the aerial-litter study.

TABLE XI  
AVERAGE PRECISION FOR THE R-CNN AND RPN USING UPSCALING OR SUPERRESOLUTION, A SUBSET OF RESULTS

Method	Mouse	Phone	T.box	Jar
Faster R-CNN	57.7	14.3	8.1	3.1
RPN and upscaling	56.8	16.4	23.4	4.1
RPN and superresolution	60.1	16.9	12.8	4.7

The litter dataset of 700 instances is close to the Average precision obtained for the tissue box at 100 instances. Krishna et al [3] mentions that 6000 instances are deemed a very small dataset that limits the ability to fine-tune the RCNN. To denote that our bi-linear upscaling of the image does not improve the average precision of either test or geoscience aerial-litter dataset. The litter detection smaller average precision number can be attributed to the foreground vs unlimited background, whereas this result is object vs object classification average precision.

*Traffic Signs:* Zhu et al [4] build an end-to-end CNN to detect and classify traffic signs. The signs. They report improvements in accuracy when data augmented with rotations between  $-20^\circ$  and  $20^\circ$ . Something which we failed to see in our study. The dataset is based on 100000 scenes with 10,000 scenes containing 30,000 traffic signs, with images subdivided into the COCO style image size categorisation of small (areas  $< 32^2$ ), medium (areas between  $32^2$  and  $96^2$ ) and large (area greater than  $96^2$ ). They perform an easy filter by cropping 25% top and 25% bottom of the image since they presume no traffic signs will be located there. The comparison cannot be performed directly as the study relies only on recall values. They quote recall values of 0.24 for small objects, 0.74 for medium objects and 0.86 for large objects. In comparison, our RCNN fares very well with these values, as our recall values range from 0.82 to 0.9 for the datasets. Meng et al [5] which use a similar sliding window approach to detect traffic signs. The dataset is based on 45 classes with more than 100 instances. The same size subsetting has been performed as Zhu et al[4]. The same recall-accuracy curves are obtained.

The Feature Pyramid Network for small object detection by Lin et al [6] is trained on COCO dataset trainval35k and tested on the minival set. The dataset contains 80k training and 35k validation, segmented. The architecture uses a feature pyramid network for scale invariancy, using Fast RCNN approach with modified Resnet-50. We obtain results on par of their small objects ( $< 32$ ) for our objects which are to be considered medium. Again the

TABLE XII  
AVERAGE PRECISION RESULTS ON COCO MINIVAL SET

Fast R-CNN	AP_{s}	AP_{m}	AP_{l}
FPN	0.178	0.377	0.458
based on conv5	0.119	0.324	0.434
based on conv4	0.157	0.365	0.455

foreground/background terrain problem does not appear to be present within this dataset. The results are shown in the table XII.

Atrous filter proposal by Guan et al [7] does not specify the PASCAL VOC 2007 benchmark with a dataset being a mix of 16.5k images obtained from VOC2007 and VOC2012. The ARPN network uses a pyramid of filters to obtain different receptive fields. The AveragePrecision they quote is **0.114 for bottles** and 0.139 for plants, which is well on the par with our results.

Eggert et al [8] on a dataset of logos from FlickrLogos dataset. The paper proposes new anchors setups and RPN on standard VGG16 architecture, and then combine to an integrated detection pipeline. The FlickrLogos images within the dataset are distinct images of man-made bold illustration on backgrounds that enhance visibility. The results are as per table ??, show very high mean average precision values as a result. Our scene background is to the detriment of object visibility rather than enhancement.

TABLE XIII  
EGGERT ET AL [8] PERFORMANCE EVALUATION ON FLICKRLOGOS DATASET

Configuration	RPN MABO	CLS (mAP)
RPN performance	0.52	0.51
RPN with original anchor sets	0.66	0.62
RPN with new anchor sets	0.68	0.66
R PN with multiple feature map and sets	0.69	0.66

Another performance with a standard background is provided by the license plate recognition by Kim et al [9]. The project uses DCNN which detects the presence of a number plate, the digits and the numbers within the same CNN architecture. The dataset was based on 200k positive images and 200k negative gray-scale images. DCNN achieves 0.993377 accuracy for license plate and 0.980266 for candidate regions.

The Perceptual GAN for small object detection by Li et al [10] also attempt the traffic sign detection. Using a perceptual generative adversarial approach with the generative branch based on a VGG16 network creates a super-resolved representation for the traffic sign. The discriminator network is a two branch network an adversarial branch to differentiate between the super-resolved feature and the original image, and the perception branch to define accuracy. The architecture was tested on the PASCAL VOC2007 achieving an **AP of 0.694, 0.602, 0.579 and 0.418** for boat, bottle, chair, and plant, respectively. This is much higher than our test results however the comparison of a multi-terrain imagery is still to be tested.

#### N. Comparison with Aerial Imagery Object Detection

Literature review of small objects use a disparate methods of evaluation. Accuracy and background accuracy, total recall and total precision are used, so direct comparisons are problematic. Official datasets such as the VEDAI dataset [11] where comparative results are published, there is a higher tendency to use official metrics such as AP. The background accuracy is a misleading accuracy as the number of tiles marked as background in large images is a large number. A very high accuracy rating would still be greater than the number of foreground annotations in an image with a very high foreground to background ratio.

1) *Detection in Urban Environments:* Detection in urban environments make use of apriori knowledge of the urban background scenarios. Vehicle detection algorithms evaluated below use this knowledge to reduce the available search space. Methods replace RPN from Faster R-CNN to accomodate the distinguishing features of

urban vs foreground images.

Deng et al [12] develop the AVPN for this purpose, using the Munich Vehicle dataset. AVPN selects from the image the best location where to look for a vehicle. Knowing that vehicles can only be located on roads and drive-ways, the precision rate is as high as 0.8593 for AVPN basic and 0.9198 for AVPNbasic + fast R-CNN. The recall rate is on par with our results of 0.7473 to 0.7702. At an IoU value of 0.2, the average precision obtained is 0.65 to 0.8 on sample of ten test images. The background/foreground discrimination ability of the algorithm on this dataset is the determining factor why it achieves this AP.

The segment before detect approach as used by Audebert et al [13] use a SegNet a Segnet that produces semantic maps. The eroded regions obtained from the maps are sent to a vehicle detecting network based on LeNet, AlexNet and VGG-16. Regions of a default minimum size (as defined by the vehicle sizes seen) are removed. The author specifically remarks the improvement of evaluation scores when removing the small region areas. The data obtained from the VEDAI dataset [11] containing detectable object of size  $30 \times 30$ . Various data augmentation techniques are performed during training. Evaluation datasets used are VEDAI [14], which are 1268 scenes ( $1024 \times 1024$  pixels), ISPRS Potsdam 38 scenes ( $6000 \times 6000$  pixels), and NZAM Christchurch 4 images ( $5000 \times 4000$  size). Precision and recall results are obtained in table XIV.

TABLE XIV  
VEHICLE DETECTION RESULTS ON THE ISPRS POTSDAM AND NZAM/ONERA DATASETS

Dataset	Precision	Recall
NZAM/ONERA	0.833	0.791
Christchurch		
ISPRS Potsdam	0.907	0.791

Zhong et al [15] also use the VEDAI dataset [11] in their VPN implementation replacing the Faster R-CNN RPN filter. VPN uses multiple feature maps instead of just the final one (like RPN) and proposes areas where vehicles might be located. The network is based on a VGG-16 network. Results and comparison to other methods is shown in table XV.

TABLE XV  
RESULTS OBTAINED BY ZHONG ET AL [15] USING A VPN NETWORK PROPOSAL

Detection Model	Image Size	Recall	AP	F1-Score
Faster R-CNN (Z&F)	1024 x 1024	0.635	0.308	.0229
Faster R-CNN (VGG-16)	1024 x 1024	0.739	0.421	0.232
Fast R-CNN (VGG-16)	1024 x 1024	0.722	0.398	0.216
SLIC with Z&F	1024 x 1024	0.583	0.254	0.066
SLIC with VGG-16	1024 x 1024	0.588	0.232	0.064
Our Model	1024 x 1024	0.723	0.546	0.320
Faster R-CNN (Z&F)	512 x 512	0.609	0.320	0.212
Faster R-CNN (VGG16)	512 x 512	0.714	0.409	0.225
Fast R-CNN(VGG16)	512 x 512	0.694	0.373	0.224
Our Model	512 x 512	0.697	0.502	0.308

The table [15] shows that a Faster R-CNN used within this type of application achieves an AP of 0.308. With such a comparably small recall one should note that the AP is being augmented primarily by the precision obtained. The task at hand shows that the RCNN has high discriminatory features that enhances the vehicle detection accuracies obtained.

Ammour et al [16] using the same UAV surveying technology for vehicle detection, uses mean-shift algorithm to segment the image into regions of the same HSV value. The classifier is based on a VGG16 architecture with an SVM classifier. Again a selection of regions is performed depending on size, with operations on the resultant mask to fill small holes, mask dilation and inspecting isolated regions. The regions were tuned by Meanshift to cover roughly one-tenth of the car, which would cover the different car features such as windscreen, roof, backscreen, side doors, etc... The results shown in table XVI.

Different to vehicle detection but still using the urban environment distinguishing features is the Solar Photovoltaic Array Detection as studied by Malof et al[17]. The algorithm uses an ensemble of Random Forest and

TABLE XVI  
AMMOUR ET AL [16] DETECTION RESULTS FOR VEHICLE DETECTION USING MEAN-SHIFT

Test Image	Cars Present	TP	FP	Recall	Precision
Image 1	56	47	4	0.839	0.922
Image 2	31	26	10	0.839	0.722
Image 3	19	14	7	0.737	0.667
Image 4	16	13	0	0.813	1.000
Image 5	5	4	2	0.800	0.667

CNN algorithms. The RF prescreens the object locations and CNN detect the PV panels. The input image is to the CNN is a  $40 \times 40$  image. The algorithm achieves a recall of 0.9. The precision recall curve is shown in figure I-N1.

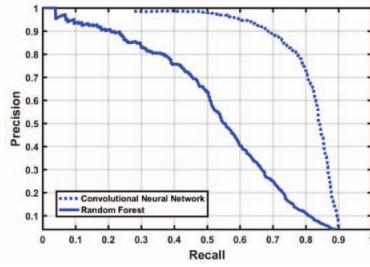


Fig. 26. Precision Recall curve of RF/CNN panel detection by Malof et al [17]

2) *Other Aerial use of CNN:* Other Aerial imagery applications do not have the urban environment of the vehicle detection, however most studies fail to mention the problematics of the high variable in the background. One should deduce that the studies did not encounter problems with the variable background.

The fast animal UAV detection by Kellenberger [18] uses 654 images from the SAVMAP project, from the Kuzikus Wildlife reserve park in Namibia. Kellenberger reduces the IoU threshold to 0.25 (as for all small object detection methods). The pre-trained Alexnet used is trained on animals against background. Improvements on the CNN is a two-branch approach that also calculates the size of the objects detected as a classifying feature. No mention of the type of backgrounds in the images is mentioned. Results shown in table XVII.

TABLE XVII  
RESULTS FOR FAST UAV DETECTION OF ANIMALS BY KELLENBERGER ET AL [18]

	Fast R-CNN (baseline)	Proposed Model
Ground-Truth Objects	509	509
True Positives	429	379
False Positives	843	254
False Negatives	80	130
Precision	0.34	0.60
Recall	0.84	0.74
F1-Score	0.48	0.66

Another aerial imagery use of CNN algorithms in ecological projects, is the detection of white geese by Bowley et al [19]. The study identifies the problematics of background variability, and stresses the importance the use of a lot of data. In fact Bowley et al [19] use a dataset of 65,000 annotated images. Very high altitudes were used, for geese detection, using a 16Mpixel camera. The CNN used an image detection of  $18 \times 18$  pixel, since the end objects were of 14 - 18 pixels in width/height. A custom CNN used with 2 convolutional layers and 2 fully connected layers. Training was done of white geese versus background. The results obtained are shown in table XVIII.

Matched observations are the joint annotations done by both experts and citizens. An error rate of 0.9995 on a  $3000 \times 2000$  image creates 9 False Positive per scene, which is not such a distant metric than that obtained by the aerial litter detection method.

Avalanche search and rescue project for real-time object detection by a UAV, is an CNN application studied by Begija et al [20] uses a pretrained GoogleNet [21] but with an SVM classifier. A sliding window approach

TABLE XVIII  
TESTING ACCURACY RESULTS OBTAINED BY BOWLEY ET AL [19]

CNN	White Phase (geese)	Background	Total
Expert	0.9626	0.9991	0.9981
Matched	0.9152	0.9995	0.9976
Unmatched	0.9526	0.9975	0.9964

is used on  $224 \times 224$  tiles. The results report is in the form of  $P_{FA}$  (probability of negative samples) and  $P_{TP}$  (probability of positive samples). This can be misleading because the  $P_{FA}$  against the whole background can result in a significant amount of tiles. The results are shown in table XIX.

TABLE XIX  
CLASSIFICATION RESULTS OF AVALANCHE SEARCH AND RESCUE BY BEJIGA ET AL [20]

	Accuracy	$P_{FA}$	$P_{TP}$
Downsample of whole image to $224 \times 224$	0.6571	0.5283	0.8462
Downsample of $672 \times 448$ tiles to $224 \times 224$	0.9429	0.1095	0.6346
Sliding Window	0.9759	0.0152	0.8065

#### O. Evaluation Conclusion

Comparing the aerial-litter detection results from the results obtained in the literature we find that the recall values are on the par with many papers, and some exceeding the detections available. However high recall with no precision is useless. A problem with our evaluation is our imposed limitation with respect to object we required identification. The CNN is able to find many more litter instances such as boots, paper/cartoon, plastic bags, cloth, etc... that would have increased the Average Precision values obtained.

When comparing the algorithm to a generic small object detection such as that of Krishna et al [3] we find that the RCNN has the same range of average precision values. Traffic signs, vehicles in an urban environment, logos etc... have a high discriminatory features between foreground and background, that makes the evaluation against these problems not comparable. The segmentation ability of the background within this vehicle detection evaluation is a clear marker of precision values obtained with a known background filter, such as a segmentation CNN pre-filter. This tallies with our pre-filter which achieved an increase in precision values within both of the datasets. However, the removal of specific backgrounds is impossible with the aerial-litter detection, as litter can be found anywhere on land. Therefore no land or land-type can be removed without reducing the recall values obtained. Vehicle detection and solar panels have another advantage of being highly homogenous in size, shape and colour.

A common aspect of the evaluation papers is the availability of large annotated datasets and default evaluation methods. In the review of these papers, the evaluation vary a lot making comparisons between them hard enough. We used a default PASCAL VOC 2012 evaluation criter. The size of our dataset is small due to the time constraints that limited data gathering and annotation. As can be seen from our tests whenever more raw data was included in the dataset our Average Precision got better. Data Augmentation is not a substitute for raw data as the iterations and permutations found in real-world imagery cannot be replicated using mathematical transformation.

## II. FUTURE WORKS

- 1) More Data - Increase terrains and sensors
- 2) Embedded CNN - GAP for proposals (image shape and size) + detector
- 3) Use of Shallower layers in the GAP VGG architecture
- 4) Custom-made CNN with learning from scratch -
- 5) Custom made with one size input layer based on image size
- 6) Automatic Object size Detection
- 7) Automatic dataset generation using Human Confirmation

## APPENDIX

## A. Mobilenetv3 Training and Testing on Dataset v3

## REFERENCES

- [1] G. Thung, "Dataset of images of trash," 2017. [Online]. Available: <https://github.com/garythung/trashnet>
- [2] N. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "{SMOTE}: {S}ynthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
- [3] H. Krishna and C. V. Jawahar, "Improving Small Object Detection." [Online]. Available: <http://cdn.iit.ac.in/cdn/cvit.iit.ac.in/images/ConferencePapers/2017/Improving-SmallObject-Detection.pdf>
- [4] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-Sign Detection and Classification in the Wild," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2110–2118, 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7780601/>
- [5] Z. Meng, X. Fan, X. Chen, M. Chen, and Y. Tong, "Detecting small signs from large images," *Proceedings - 2017 IEEE International Conference on Information Reuse and Integration, IRI 2017*, vol. 2017-Janua, pp. 217–224, 2017.
- [6] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 936–944, 2017.
- [7] T. Guan and H. Zhu, "Atrous Faster R-CNN for Small Scale Object Detection," *Proceedings - 2017 2nd International Conference on Multimedia and Image Processing, ICMIP 2017*, vol. 2017-Janua, pp. 16–21, 2017.
- [8] R. L. Christian Eggert , Stephan Brehm , Anton Winschel , Dan Zecha, "A CLOSER LOOK : SMALL OBJECT DETECTION IN FASTER R-CNN Multimedia Computing and Computer Vision Lab," in *Proceedings of the 2017 International Conference on Multimedia and Expo*, vol. 0, University of Augsburg, 2017.
- [9] H.-H. Kim, J.-K. Park, J.-H. Oh, and D.-J. Kang, "Multi-task convolutional neural network system for license plate recognition," *International Journal of Control, Automation and Systems*, vol. 15, no. 6, pp. 2942–2949, 2017. [Online]. Available: <http://link.springer.com/10.1007/s12555-016-0332-z>
- [10] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual Generative Adversarial Networks for Small Object Detection," pp. 1951–1959, 2017. [Online]. Available: <http://arxiv.org/abs/1706.05274>
- [11] S. Razakarivony and F. Jurie, "Vehicle Detection in Aerial Imagery : A Small Target Detection Benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2015.
- [12] Z. Deng, H. Sun, S. Zhou, J. Zhao, and H. Zou, "Toward Fast and Accurate Vehicle Detection in Aerial Images Using Coupled Region-Based Convolutional Neural Networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 8, pp. 3652–3664, 2017.
- [13] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images," *Remote Sensing*, vol. 9, no. 4, 2017.
- [14] S. Razakarivony and F. Jurie, "Vehicle Detection in Aerial Imagery : A Small Target Detection Benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203, 2015.
- [15] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random Erasing Data Augmentation," 2017. [Online]. Available: <http://arxiv.org/abs/1708.04896>
- [16] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sensing*, vol. 9, no. 4, 2017.
- [17] J. M. Malof, K. Bradbury, L. M. Collins, and R. G. Newell, "A Deep Convolutional Neural Network and a Random Forest Classifier for Solar Photovoltaic Array Detection in Aerial Imagery," *International Conference on Renewable Energy Research and Applications (ICRERA)*, vol. 5, pp. 650–654, 2016.
- [18] B. Kellenberger, M. Volpi, and D. Tuia, "Fast animal detection in UAV images using convolutional neural networks," *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2017-July, pp. 866–869, 2017.
- [19] C. Bowley, M. Mattingly, A. Barnas, S. Ellis-Felege, and T. Desell, "Toward using citizen scientists to drive automated ecological object detection in aerial imagery," *Proceedings - 13th IEEE International Conference on eScience, eScience 2017*, pp. 99–108, 2017.
- [20] M. B. Bejiga, A. Zeggada, and F. Melgani, "Convolutional neural networks for near real-time object detection from UAV imagery in avalanche search and rescue operations," *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 693–696, 2016.
- [21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2015. [Online]. Available: <http://arxiv.org/abs/1512.00567>