R-FCN Object Detection Ensemble based on Object Resolution and Image Quality

Christoffer Bøgelund Rasmussen, Kamal Nasrollahi and Thomas B. Moeslund
Visual Analysis of People (VAP) Laboratory, Aalborg University, Aalborg, Denmark

Keywords: Convolutional Neural Networks, Object Detection, Image Quality Assessment, Ensemble Learning.

Abstract: Object detection can be difficult due to challenges such as variations in objects both inter- and intra-class. Additionally, variations can also be present between images. Based on this, research was conducted into creating an ensemble of Region-based Fully Convolutional Networks (R-FCN) object detectors. Ensemble strategies explored were firstly data sampling and selection and secondly combination strategies. Data sampling and selection aimed to create different subsets of data with respect to object size and image quality such that expert R-FCN ensemble members could be trained. Two combination strategies were explored for combining the individual member detections into an ensemble result, namely average and a weighted average. R-FCNs were trained and tested on the PASCAL VOC benchmark object detection dataset. Results proved positive with an increase in Average Precision (AP), compared to state-of-the-art similar systems, when ensemble members were combined appropriately.

1 INTRODUCTION

Object detection is a fundamental area of computer vision that has had a great amount of research over the past decades. The general goal of object detection is to find a specific object in an image. The specific object is typically from a pre-defined list of categories that are of interest. Object detection generally consists of two larger tasks; localisation and classification. Localisation is typically drawing a bounding-box around the object indicating where a given object is in the image and classification is determining the type of the object with an associated confidence.

Object detection is a challenging problem due to both large scale issues and minute differences between objects. Firstly, there is the challenge of differentiating objects between classes. Depending on the problem at hand the number of potential classes present can be thousands or tens of thousand. On top of this, separate object categories can be both very different in appearance, for example an apple and an aeroplane, but separate categories can also be similar in appearance, such as dogs and wolves. These main challenges of object detection stem from two categories which defined per (Zhang et al., 2013) as: robustness-related and computational-complexitity and scalability-related.

Robustness-related refers to the challenges in appearance variations within the both of intra-class and inter-class. These variations can be categorised into two types as per (Schroff, 2009) as: object and image variations. Object variations consist of appearance differences between object instances with respect to factors such as colour, texture, shape, and size. Image variations are differences not related to the object instances themselves but rather the actual image. This can consist of conditions such as lighting, viewpoint, scale, occlusion, and clutter. Based upon these differences the task of both classifying a given object as a given class but also differentiating the potentially largely varying objects into the same class is challenging.

Current state-of-the-art in object detection is within the realm of deep learning with Convolutional Neural Networks (CNN)s. Deep learning methods are of such a scale that given appropriate data have been able to address the two main challenges mentioned earlier. This is exemplified with almost all leading entries in benchmark challenges such as PASCAL VOC (Everingham et al., 2010), ImageNet (Russakovsky et al., 2015), and MSCOCO (Lin et al., 2014) consisting of CNN-based approaches. Additionally, recent trends with CNN-based object detection methods have been to incorporate ensembles of networks to further enhance performance (He et al., 2015) (Huang et al., 2016) (Li et al., 2016).

Rasmussen C., Nasrollahi K. and Moeslund T.
R-FCN Object Detection Ensemble based on Object Resolution and Image Quality.
DOI: 10.5220/0006511301100120
In *Proceedings of the 9th International Joint Conference on Computational Intelligence (IJCCI 2017)*, pages 110-120
ISBN: 978-989-758-274-5

One of the main goals of an ensemble system is to reduce the variance incorporated in the training process. An example is to train classifiers on different subsets of the data, creating a number of different ensemble members. The assumption is that the classifiers will make different errors on a given data point. However, by combining the classifiers the errors will be mitigated by the increased strength from lower individual variance. The ensemble members created in this work address two of the three main strategies from (Zhang and Ma, 2012) to build an ensemble system. Namely:

1. Data sampling and selection: selection of training data for individual classifiers.
2. Training member classifiers: specific procedure used for generating ensemble members.
3. Combining ensemble members: combination rule for obtaining ensemble decision.

The robustness-related challenges are addressed by exploring the possibilities of designing expert ensemble members towards both object and image variations in a leading object detection benchmark. This is done by training an ensemble of Region-based Fully Convolutional Network (R-FCN) with ResNet-101 networks. Data sampling strategies are used to create subsets of data with respect to object resolution and various image quality factors. Finally, two separate combination strategies are explored for combining the ensemble members. The rest of this paper is organized as follows: the related works are reviewed in the next section.

2 RELATED WORK

One of the first methods to show that CNN could significantly improve object detection was that of R-CNN (Girshick et al., 2014). The method obtains the name R-CNN as a Convolutional Neural Network (CNN) is used on regions of the image. Regions are pre-computed as proposals using a method such as SelectiveSearch (Uijlings et al., 2013) to give an indication as to where objects may be located. In R-CNN the CNN model is used as a feature extractor from which a class-specific linear Support Vector Machine (SVM) can be trained on top of. The AlexNet-based feature extractor is firstly pre-trained on a large dataset designed for classification and then fine-tuned to object detection. Each pre-computed region proposal is run through a forward pass of the model to extract features and then passed to the SVM.

The R-CNN method was improved the following year with Fast R-CNN (Girshick, 2015) and aimed to improve speed and accuracy. One of the significant changes is that training end-end rather than in the multi-stage pipeline in R-CNN. A CNN is again used as a feature extractor where Region of Interest (RoI) pooling is conducted on the final feature map. Afterwards the forward pass continues through two fully-connected layers followed by two sibling output layers replacing the external SVM. The sibling outputs are a softmax classification layer that produces probabilities for the object classes and another layer for bounding-box regression. In R-CNN, the only deep network used was AlexNet (Krizhevsky et al., 2012), however, in Fast R-CNN the authors experiment with networks of different size. It was found that the deeper network VGG-16 (Simonyan and Zisserman, 2015) for computing the convolutional feature map gave a considerable improvement in performance. As the name implies the main improvement is the speed in respect to both training and testing. By computing a convolutional feature map for an entire image rather than per object proposal the number of passes in the network is lowered significantly. While Fast R-CNN provided improvements in both accuracy and speed, the increase in speed is only in relation to the actual object detection and assumes that the region proposals are pre-computed. Therefore, there is still a significant bottleneck per image as a region proposal method can typically take a couple of seconds.

Faster R-CNN (Ren et al., 2015) addressed this bottleneck in the third iteration of the R-CNN method. Faster R-CNN showed that region proposals could be computed as part of the network through the use of a Region Proposal Network (RPN). The RPN shares the convolutional layers and feature map used for computing features with RoI pooling in Fast R-CNN. As these layers are already computed on the entire image for the classification pipeline, the added time for proposals using the RPN is negligible. Apart from the change in how region proposals are computed, there is no difference in comparison to Fast R-CNN. An RPN takes the last convolutional feature map as input and returns a number of object proposals.

The winner of the Microsoft Common Objects in Context (MS COCO) 2015 and ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015 detection challenge was based on deep residual networks (ResNets) (He et al., 2015). As is well known with CNNs, deeper networks are able to capture richer higher-level features. The authors showed that this is also beneficial in the object detection domain. In (He et al., 2015) an ensemble of three deep residual networks with 101 layers was trained for object detection and another ensemble of three used for region proposals with the RPN while being based on the Faster

R-CNN framework. In addition to the ensemble, the winning entry also added box refinement, global context, and multi-scale testing to the Faster R-CNN.

The current leading method on MS COCO is an extension of the previously explained ResNets (He et al., 2015). This method denoted as G-RMI on the MS COCO leaderboard (COCO, 2017) is an ensemble of five deep residual networks based upon ResNet (He et al., 2015) and Inception ResNet (Szegedy et al., 2016) feature extractors. No work has been published yet on G-RMI at this time, however, a short explanation of the entry is included in a survey paper from the winning authors (Huang et al., 2016). The approach was to train a large number of Faster R-CNN models with varying output stride, variations on the loss function, and different ordering of the training data. Based upon the collection of models, five were greedily chosen based upon performance on a validation set. While performance on the models were important, the models were also chosen such that they were not too similar.

Recently, a newer approach to region-based methods has been proposed with the use of Fully Convolutional Networks (FCNs) through the R-FCN (Dai et al., 2016). The overall approach is similar to that used in region-based methods such as (Girshick et al., 2014), (Girshick, 2015) and (Ren et al., 2015). First, it computes region proposals using a region proposal method and then it performs a classification on these regions. R-FCN uses the RPN from Faster R-CNN (Ren et al., 2015) for proposal computation. However, RoI pooling is performed on position sensitive score maps rather than the last feature map. The score maps are split up to represent a relative position in a $k \times k$ grid, with each cell presenting information relative to the spatial position of an object.

3 OBJECT DETECTION WITH R-FCN

One of the current leading object detection methods is the R-FCN (Dai et al., 2016). The authors of R-FCN were inspired by the recent advances in FCN classification networks. R-FCN uses position-sensitive score maps computed by a bank of convolutional layers. The maps add translation variance into the detection pipeline by computing scores in relation to position information with respect to the relative spatial position of an object. A RoI-pooling layer is added after the score-maps, however, no convolutional operations are done after this point ensuring translation variance.

The overall approach of the R-FCN also consists of the popular two-stages of region proposal and region classification. Region proposal is done using the RPN from Faster R-CNN followed by the position-sensitive score maps and RoI pooling for region classification. Similar to Faster R-CNN, convolutional layers are applied on the input image and the RPN computes region proposals. After this, position-sensitive score maps aid in classification.

The added translation variance post finding proposals with the RPN by producing a bank of k^2 score maps for each object category. Therefore, there are a total of $k^2(C+1)$ maps. The number of k^2 maps is due to a $k \times k$ spatial grid representing relative positions. Typically $k = 3$, therefore, nine score maps represent position-sensitive scores for a given object category. For a given RoI placement the vote for relative position is sampled from their respective map in the bank.

Once the bank of score maps have been computed, position-sensitive RoI-pooling is found for region classification. Each individual $k \times k$ bin pools from its corresponding location in the relevant score map. For example, the top left bin pools from that position in the top-left score map and so on. The final decision for a given class is determined by a vote where each of the bins are averaged, producing a $(C+1)$-dimensional vector for each RoI.

4 PROPOSED METHOD

An ensemble of R-FCNs with the ResNet-101 model will be trained towards different robustness-related challenges in the Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes (PASCAL VOC) dataset. The data used will follow the leading methods for PASCAL VOC 2007 object detection. Training will be done on the 07+12 train sets and testing was conducted on the 07 test set. Evaluation was conducted using the Average Precision (AP) metric as per the 07 guidelines.

Leading object detection systems take advantage of ensemble methods. Many of them are trained with regards to the variations in internal architecture and not specifically training experts towards solving specific challenges. Therefore, the system in this work will take advantage of the first ensemble strategy from (Zhang et al., 2013), data sampling and selection. The individual R-FCNs were trained on different subsets of training data with the aim to create expert ensemble members in regards robustness-related challenges, namely object resolution and image quality.

The third strategy in building an ensemble system, to combine predictions from individual members of the ensemble is also addressed. Bounding-boxes and the confidence of each detection will be combined using an averaging and a weighted averaging method is tested on a number of different combinations of ensemble members.

4.1 Training Ensemble Members

The training of the R-FCN members will be done using Convolutional Architecture for Fast Feature Embedding (Caffe) (Jia et al., 2014). This was chosen due to the research being available from the authors of R-FCN through training code and pre-trained Caffe models. As there is the requirement to combine detections between ensemble members, the detections must be found based upon the same input to each model. This is ensured by using pre-computed region proposals found using an RPN. In a standard R-FCN the RPN is an internal part of the network and is trained end-to-end. However, as these proposals must be constant between all ensemble members this method is not appropriate. Instead the networks are trained using a method inspired by the 4-step alternating training method presented by the Faster R-CNN authors (Ren et al., 2015). The process can be seen in Figure 1.

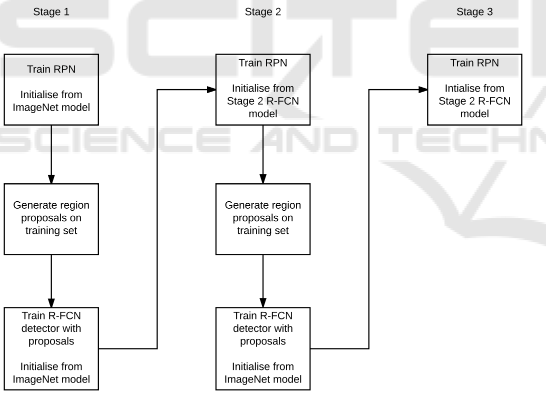

Stage 1
Stage 2
Stage 3
Train RPN
Initialise from ImageNet model
Train RPN
Initialise from Stage 2 R-FCN model
Train RPN
Intialise from Stage 2 R-FCN model
Generate region proposals on training set
Generate region proposals on training set
Train R-FCN detector with proposals
Initialise from ImageNet model
Train R-FCN detector with proposals
Initialise from ImageNet model

Figure 1: Flow chart showing the alternating training method.

In this approach the overall network in trained in multiple steps. First, an RPN is trained to determine region proposals, the RPN is initialised from a pre-trained ImageNet model and fine-tuned to the proposal task. Next a R-FCN is trained based upon the proposals found in the previous step. This network is also initialised with a pre-trained ImageNet model. In step three, another RPN is trained but initialised using the R-FCN from step two. In this step the convolutional layers that are shared between the R-FCN and RPN are fixed and only the layers unique to the RPN are updated. By training a model with this approach a testing image is able to run through the same steps as a R-FCN trained end-to-end, however, as the networks are split into different models it is also possible to use the stages of the method individually. Creating a solution for finding region proposals with an external RPN and having a number of R-FCNs that can take the proposals as inputs.

An additional benefit to training R-FCNs in this manner is that once a baseline model has been created only one part needs to be re-trained. As the aim is to train various ensemble members to different subsets of data only the R-FCN in stage 2 is required to be re-purposed. The RPN in stage 3 should be kept constant based on the baseline model as it will provide the shared proposals for test images. Therefore, once a systematic approach has been found for splitting data for both train and test based on the data sampling and selection requirements the detection part of the R-FCN can be trained towards its expert area. The following sections will explain how the subsets of data will be selected.

4.1.1 Object Size Data Sampling

The area of a region proposal found with a RPN gives an indication as to the approximate size of a potential objects. Therefore, the area for all proposals on the training set can be computed from the output of the second step in stage 2 shown in Figure 1. Once the area of all proposals are computed an appropriate split of the data can be determined depending on the area distribution. The main requirement in creating the subsets of data is that equal number of ground truth samples should be present in both.

4.1.2 Image Quality Data Sampling

There are many choices for computing the quality of an image and a popular area of research for this purpose is Image Quality Assessment (IQA). These methods aim to determine the subjective quality of an image. There are two forms of IQA, Full-Reference Image Quality Assessment (FR-IQA) and No-Reference Image Quality Assessment (NR-IQA). FR-IQA approaches require the original, undistorted reference image in order to determine quality. Whereas, NR-IQA do not have this information available (Bosse et al., 2016). As the aim is to determine the level of image quality on one of the benchmark datasets, no reference image is present. Therefore, an NR-IQA method is required. Current state-of-the-art within NR-IQA is also deep learning based and works are typically trained on IQA datasets. Datasets include Laboratory for Image & Video Engineering (LIVE) dataset (Sheikh et al., 2006) (Sheikh et al.,), TID2013 (Ponomarenko et al.,

2013) and CSIQ (Larson and Chandler, 2009). The datasets consist of source reference image and have artificially created counterparts with varying levels of distortion. Distortions include, such as in the LIVE dataset, JPEG2000 compression, JPEG compression, additive white Gaussian noise, Gaussian blur and bit errors from a fast fading Rayleigh channel. Models can then be trained to predict subjective quality based on ground truth user determined quality measurement.

Based upon this, an NR-IQA method can be used to determine the level of image quality with respect to a number of different distortions. Then as in object size training the data will be split into appropriate training subsets.

4.1.3 R-FCN Training

Training of the baseline R-FCN model shown in Figure 1 is done using Stochastic Gradient Descent (SGD) optimisation with largely the same parameters across the five different training parts. The parameters are adapted from (Dai et al., 2016). All models start with a base learning rate of 0.001 which is dropped by a factor of 0.1 once in the process. This is done after 80,000 iterations for the R-FCN models and after 60,000 for the RPNs. The learning rate is controlled with a momentum of 0.9 and weight decay of 0.0005. The two R-FCN models are trained for 120,000 iterations, while the three RPNs are trained for 80,000. The only data augmentation used in training is horizontal flipping of images, effectively creating double the amount of training examples. Additionally, Online Hard Example Mining (OHEM) (Shrivastava et al., 2016) is used in the training process.

5 RESOLUTION-AWARE ENSEMBLE MEMBERS

To determine an appropriate split of data the distribution of the ground truth bounding boxes area from the 07+12 set was analysed. This was done by parsing all of the bounding box coordinates in the set and calculating the area. A histogram of the all of the ground truth areas can be seen in Figure 2. There is a clear tendency to smaller objects in the training set with a clear skew towards the left of the figure. The data in Figure 2 can be split into two equal subsets if the median area of 19,205.5 is used as indicated by the red line.

However as mentioned, the ensemble R-FCN members are trained with region proposal inputs of both ground truth positives and negative examples found using a RPN as per the multiple step training scheme. A potential shortcoming of using proposals as inputs to training ensemble members is a RPN finds many more examples of possible objects than actually are present. Ground truths are determined by setting the proposals with the highest confidence as the ground truth examples and labelling the remaining proposals as the background class. This creates a large difference in the number of samples in comparison to the end-to-end training approach. The total number of training examples is increased from 80,116 ground truth object instances to 9,979,345 region proposals. The median of the almost 10 million proposals is 4,684 pixels, a significantly less than 19,205.5 determined using only ground truth boxes. If the subsets were split by the median of all RPN proposals (4,684), the two sets of data would have equal numbers of examples. However, there appears to be a large skew in RPN proposals to smaller objects and therefore there significantly more ground truth samples in the subset of data containing larger objects. This can be seen in Table 1, where despite there being an almost even split in data subsets there are significantly more ground truth annotations in the RPN$_{larger}$ subset.

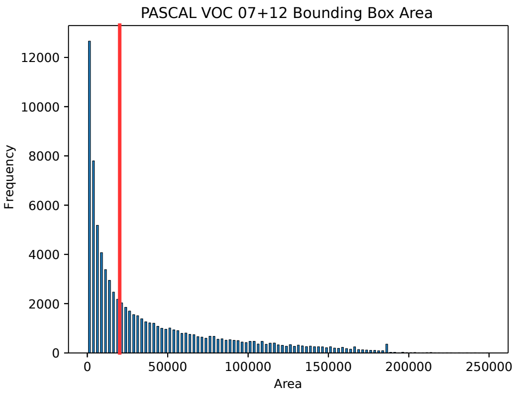


Figure 2: Histogram of the PASCAL VOC 07+12 bounding box area.

Table 1: Creating object resolution data subsets. If split by the median area of all region proposals training samples the larger dataset has significantly more ground truth object instance samples.

Data	**RPN$_{smaller}$**	**RPN$_{larger}$**
Ground Truth	19,992	60,116
Background	4,969,369	4,929,297
Total	4,989,361	4,989,413

Another option is to use the median of 19,205.5 found on only ground truth boxes. The data distribution based on this threshold can be seen in Table 2. In this instance there is significantly more data in the

Table 2: Creating object resolution data subsets. If split by the median of area from ground truth objects there is an equal number of ground truth instances. However, RPN$_{larger}$ has significantly more background samples.

Data	**RPN$_{smaller}$**	**RPN$_{larger}$**
Ground Truth	40,058	40,058
Background	3,528,370	6,370,859
Total	3,568,428	6,410,917

RPN$_{larger}$ subset, however, the skew is solely due to the many more background examples. The ground truth annotations are shared equally with 40,058 samples in each.

As the overall goal of object detectors is to find objects within the classes, the decision was made to use the threshold of 19,205.5 to create the split in data, despite there being significantly more background examples in one of the datasets.

The R-FCN ensemble members were trained on the two subsets of RPN. To evaluate how well the expert resolution members perform on the respective subsets of data tests were performed on splits of the 07 test data. This data was split by using the same median threshold of 19,205.5 used in creating the training subsets. Firstly, the results for small objects from 07 test can be seen in Table 3. Shown are R-FCNs trained on RPN$_{smaller}$, RPN$_{larger}$ and a baseline model trained on all 07+12 data. The table shows that the model trained towards smaller object proposals on RPN$_{smaller}$ performs best. This trend is similarly true for large objects as seen in Table 4. Finally, for all ground truth objects the baseline model is the best performing as seen in Table 5.

Table 3: Results for R-FCN models trained on three different subsets of data and tested on only small objects from the 07 test set.

Train Data	**AP**
RPN$_{smaller}$	**55.00**
RPN$_{larger}$	20.92
07+12	43.80

Table 4: Results for R-FCN models trained on three different subsets of data and tested on only large objects from the 07 test set.

Train Data	**AP**
RPN$_{smaller}$	21.28
RPN$_{larger}$	**81.81**
07+12	75.14

5.1 Image Quality Ensemble Members

Table 5: Results for R-FCN models trained on three different subsets of data and tested on all of the 07 test set.

Train Data	**AP**
RPN$_{smaller}$	46.74
RPN$_{larger}$	62.48
07+12	**79.59**

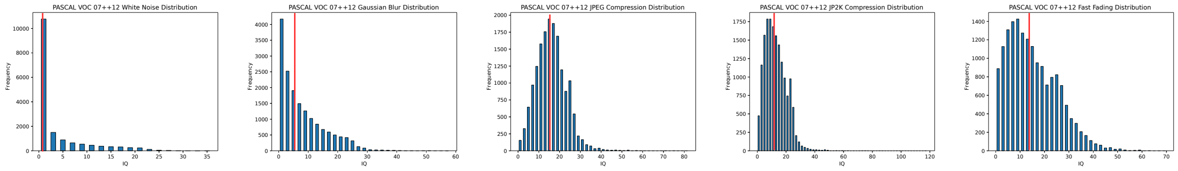
To evaluate the amount of distortion in the PASCAL VOC dataset a method for IQA is needed. A recent state of the art method is that of deep IQA (Bosse et al., 2016). Deep IQA is a CNN-based No-Reference (NR) IQA method that can be trained to measure the subjective visual quality of an image. Deep IQA consists of 14 convolutional layers, 5 max-pooling layers and 2 fully-connected layers. The convolutional layers are all 3×3 convolution kernels and activated using Rectified Linear Unit (ReLU). Inputs to each convolutional layer are zero-padded to ensure output size is equal to the input. Max-pooling layers consist of 2 × 2 sized kernels. The network is trained on mini-batches of 32 × 32 patches. During inference non-overlapping patches are sampled from the image and image quality scores are predicted for each instance. The patch scores are averaged for the final score for the entire image.

Deep IQA models were trained using the Chainer framework (Tokui et al., 2015) as code and a model trained for all distortions types on the LIVE dataset were available from the deep IQA authors. However, to create a more powerful ensemble models were fine-tuned from the model provided towards each of the 5 distortions in the LIVE. The training settings are the same as in the deep IQA work apart from the number of epochs in training. As fine-tuning can drastically decrease training time the epochs were decreased from 3,000 to 500.

The models for each distortion type are run through the 07+12 dataset in order to give an indication to the respective distributions. The distributions can been seen in the histograms in Figure 3.

The distribution for white noise and Gaussian blur is skewed towards a higher image quality and also to a lesser extent in fast fading. Whereas the image quality for compression distortions is somewhat of a Gaussian nature. For determining an appropriate manner to split the data the same constraints are made as in that for object sizes, namely that both subsets of data should have an equal number of ground truths to train on. Again using the median for each of the five distributions can satisfy this. The respective medians can be seen in Table 6 and are shown by the red lines in Figure 3.

It does not appear feasible to create subsets of data for white noise image quality on 07+12. The combination of both the heavy skew and half of the data lying below 0.599 indicates that a minimal amount



White Noise Gaussian Blur JPEG JP2K Fast Fading

Figure 3: Histograms representing the distribution of image quality for the five distortions trained from the LIVE image quality dataset. The distortions shown are white noise (a), Gaussian blur (b), JPEG compression (c), JP2k compression (d), fast fading (e).

Table 6: Median values used for each distortion type to create even subsets of training data from 07+12.

Distortion Type	**Median**
White Noise	0.599
Gaussian Blur	5.607
JPEG Compression	15.660
JP2K Compression	11.747
Fast Fading	13.373

of white noise distortion is present. Therefore, this distortion is not considered for part of the ensemble. While the Gaussian blur image quality is also skewed it is similar to that of the the object sizes and therefore is deemed appropriate to split based upon its median of 5.607. The remaining distributions are much less skewed and a total of eight R-FCN models will be trained for the high and low levels of image quality for the distortions Gaussian blur, JPEG compression, JP2K compression and fast fading. Therefore, in total there will be ten R-FCN models trained including the two for smaller and larger object sizes.

As in the resolution-aware R-FCN networks individual tests are run to evaluate whether or not the models trained on the above data are candidate experts. The 07 test set is split into lower and upper subsets for each distortion type according to their respective medians. The two respective experts trained on each of their subset and the baseline R-FCN model are evaluated on their respective subsets. Similar results are not found in this instance as in the object resolution experts. For each of the 5 distortions both of the trained experts perform very similarly and are generally 3-4 AP lower than the R-FCN model trained on all of the data. Regardless of this result the following section will present a method to ensemble these members as they may still complement each other.

6 COMBINING THE ENSEMBLE MEMBERS

The two strategies, average and weighted average, for combining the ensemble members will be described in this section. The method for inferring each test image will be the same apart from the combination step. For a given object proposal each network will infer a bounding box and associated confidence for all classes. After this the given ensemble combination method determines the final detection where the confidence and the four corners of the bounding box will be averaged.

6.1 Average Ensemble

Each of the five ensemble factors are weighted evenly in the overall ensemble. Within each ensemble factor pair, the detection for one of the pairs will be chosen and the other discarded. This is determined by where the given factor lies for the test image in relation to the training data distribution. For example, if it is measured that an image with a deep IQA model to have JPEG compression below the threshold used to split the data, then the detection found using the model trained on that data will be used. This results in five detections that will be weighted equally to find the final detection by:

$$E_j = \frac{1}{n}\sum_{i=1}^{n} p_{i,j} \tag{1}$$

where n is the number of detections found by the n ensemble factor, p is the detection result to be averaged and i represents one of the ensemble factors. Finally, j is one of the five values found by each detection, namely the four corners of the bounding-box and the associated confidence.

6.2 Weighted Average Ensemble

As in the average ensemble, each of the 10 trained networks will be used on all object proposals found using the RPN. Between factors, weights are distributed evenly across each of the five different types of factors as in the average ensemble. The weighted average ensemble is determined for each bounding box and the associated confidence by:

$$E_j = \frac{1}{n}\sum_{i=1}^{n} w_i p_{i,j} \tag{2}$$

where w_i is the weight for a given detection.Weights are determined in pairs for each of the 5 ensemble factors, where the total sum of weights is equal to n. If each detection were to be weighted equally all w would be equal to 1. As the weights are calculated in pairs each ensemble factor is overall weighted equally as the pair of weights can at most be equal to 2. By using this tactic, detections between ensemble members can be weighted differently but each factor is weighted equally. Weights for a given factor are found according to where the the test image lies for that factors training data distribution.

If the image factor result f_i, for example proposal size, is below the value used to split the data the weights are calculated for the detection found with the given lower network by:

$$w_{Lower} = 1 + \frac{median_i - f_i}{median_i - minf_i} \tag{3}$$

and the weight for the upper network w_{Upper} by:

$$w_{Upper} = 2 - w_{Lower} \tag{4}$$

where $median_i$ is the value used to split the training data and $minf_i$ is the minimum quality for the given factor in the training set.

However, if the quality is above $split$ the w_{Upper} is calculated by:

$$w_{Upper} = 2 - \frac{maxf_i - f_i}{maxf_i - median_i} \tag{5}$$

and lower weight w_{Lower}:

$$w_{Lower} = 2 - w_{Upper}. \tag{6}$$

It should also be noted that outliers are not included for the calculation of $minf_i$ and $maxf_i$ by removing the values below the 1% and above the 99% percentile. This ensures that the weighing of factors is not too heavily affected by outlier values.

7 EXPERIMENTAL RESULTS

In this section the results for the two aforementioned ensemble combinations strategies will be presented. When appropriate the result for the baseline R-FCN ResNet-101 model trained on all of the 07+12 training data and will be presented and denoted as Baseline. The results presented will be on the 07 PASCAL VOC test set as also shown in earlier preliminary results in this report.

Table 7: Results for the two ensemble combination strategies and for the baseline model on the 07 test set.

Method	AP
Average	79.45
Weighted Average	79.47
Baseline (Dai et al., 2016)	**79.59**
Faster R-CNN (He et al., 2015)	76.4
YOLOv2 (Redmon and Farhadi, 2016)	78.6

The results for both combination strategies using 10 ensemble members can be seen in Table 7.

While neither of the combinations provide an improvement over the baseline method, both have an increase in performance in comparison to their respective image quality expert results.

To the evaluate the contribution of both the eight quality factor ensemble members and the two resolution members these were combined separately based on the two strategies. By separating the quality and resolution members the performance decreases by roughly 1.0 for both in comparison the the average ensemble result. This appears to indicate that the two complement each other well and have their own expertise for this problem. The weighted average combination strategy does not show as large of a decrease in performance for image quality as the average combination does, however, there is still a drop from 79.47 to 79.04. There is also a decrease in performance for the two resolution members showing an AP of 77.84 on the test set. This seems to show that the addition of weighing individual detections based on proposal size as a poorer approach. There appears to be an indication that image quality members are well suited to adding a weight to detection. Whereas, the resolution members are better suited to simply taking the detection from the appropriate model. The results for this can be seen in Table 8 where both combinations are tested. The two strategies are shown as either Image Quality or Resolution followed by the subscript $_{Avg}$ or $_{WAvg}$ indicating the combination strategies of average or weighted average respectively.

Table 8: Results for the the image quality ensemble members and resolution members with both combinations of average and weighted average on the 07 test set.

Ensemble Members	AP
Image Quality$_{WAvg}$ / Resolution$_{Avg}$	**79.83**
Image Quality$_{Avg}$ / Resolution$_{WAvg}$	79.17
Baseline (Dai et al., 2016)	79.59
Faster R-CNN (He et al., 2015)	76.4
YOLOv2 (Redmon and Farhadi, 2016)	78.6

Results in Table 8 show that by using separate strategies where image quality members are weighted

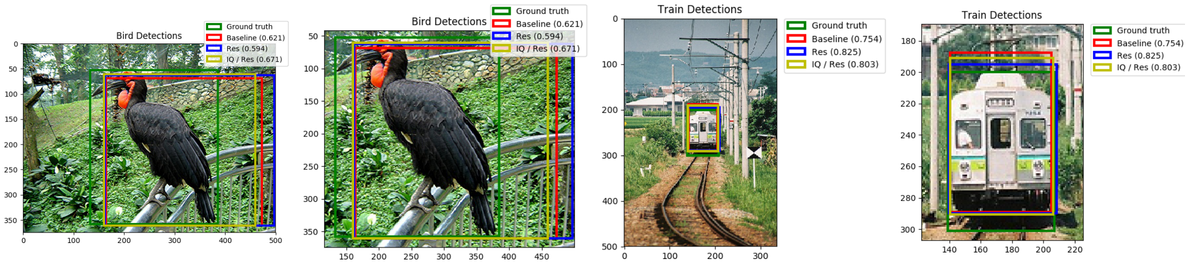


Figure 4: Detections for the bird class from an image in the 07 test set. Shown are the bounding boxes for the ground truth annotation, baseline, Resolution$_{base}$ (Res) and Image Quality$_{WAvg}$ / Resolution$_{Avgbase}$ (IQ / Res). The Intersection-Over-Union (IoU) between the ground truth and bounding box is shown in parentheses for each method.

and when resolution members are averaged only increases the performance. Additionally, the performance surpasses the baseline model.

The results so far have only been with different combinations of the expert ensemble members. Another strategy is to include the baseline model trained on all of the 07+12 data. As the baseline model performs well by itself the other ensemble members will act as support It should be noted that as there is no complementary member to the baseline. Therefore, its detections are weighted by 1.0 regardless of ensemble combination strategy. Firstly, the results for the average and weighted average ensemble can be seen in Table 9. The inclusion of the baseline model is shown by the subscript $_{base}$. Performance is increased in both cases, the weighted average is increased by 0.22. While the average strategy is increased by 0.43.

Table 9: Results for the two ensemble combination strategies and for the baseline model on the 07 test set. Shown is both the results with the expert ensemble members only and experts plus the baseline model.

Method	**AP**
Average	79.45
Average$_{base}$	**79.88**
Weighted Average	79.47
Weighted Average$_{base}$	79.69
Baseline (Dai et al., 2016)	79.59
Faster R-CNN (He et al., 2015)	76.4
YOLOv2 (Redmon and Farhadi, 2016)	78.6

The addition of the baseline model to the ensemble using different strategies for the two factors can be seen in Table 10. This provided the best result of any ensemble combination. Image quality with the weighted average and resolution with average ensemble results in 80.09, an increase of 0.50 in comparison to the baseline R-FCN.

The AP results for each category for the Image Quality$_{WAvg}$ / Resolution$_{Avgbase}$ ensemble can be seen in Table 11. The tables show results for the baseline model, the given ensemble method and the difference between the two for a given class.

Table 10: Results for the the image quality ensemble members and resolution members with both combinations of average and weighted average on the 07 test set. Shown is both the results with the expert ensemble members only and experts plus the baseline model.

Ensemble Members	**AP**
Image Quality$_{WAvg}$ / Resolution$_{Avg}$	79.83
Image Quality$_{WAvg}$ / Resolution$_{Avg\ base}$	**80.09**
Image Quality$_{Avg}$ / Resolution$_{WAvg}$	79.17
Image Quality$_{Avg}$ / Resolution$_{WAvg\ base}$	79.54
Baseline (Dai et al., 2016)	79.59
Faster R-CNN (He et al., 2015)	76.4
YOLOv2 (Redmon and Farhadi, 2016)	78.6

Finally, two examples of detections can be seen in Figure 4. For both instances, on the left is the full size image and right a zoomed version of the object and detections. The detections shown are for the ground truth annotation, baseline, Resolution$_{base}$ (Res) and Image Quality$_{WAvg}$ / Resolution$_{Avgbase}$ (IQ / Res). Additionally, shown in parentheses in the legend is the IoU between the ground truth and detection for the given method.

8 CONCLUSION AND FUTURE WORK

This work has presented a method for creating an ensemble of R-FCNs trained towards object resolution and image quality using the PASCAL VOC dataset. If combined appropriately an improvement against the standard R-FCN method can be obtained. Addressing items such as the skew in factor distributions data may help create better individual members and create a stronger ensemble.

This work uses R-FCN as the backbones, however, any object detection method could be used and shows the possibilities of engineering towards specific challenges in object detection.

Table 11: Results for the individual classes in the 07 test set. Shown are the results for the baseline model and Image Quality$_{WAvg}$ / Resolution$_{Avgbase}$. Additionally the difference between the two methods are presented for a given class.

Model	**aero**	**bike**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**
Baseline (Dai et al., 2016)	80.53	84.59	79.89	71.52	67.54	87.22	**87.59**	87.98	65.15	**87.11**
Image Quality$_{WAvg}$ / Resolution$_{Avgbase}$	**80.57**	**85.45**	**81.02**	**72.51**	**68.69**	**88.00**	87.38	**89.13**	**67.27**	86.57
Difference	+0.04	+0.86	+1.13	+0.99	+1.15	+0.78	-0.21	+1.15	+2.12	-0.54

Model	**table**	**dog**	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**
Baseline (Dai et al., 2016)	**73.66**	88.61	**87.83**	83.21	79.87	**54.60**	**84.07**	80.03	83.60	77.17
Image Quality$_{WAvg}$ / Resolution$_{Avgbase}$	72.21	**88.75**	87.04	**84.15**	**80.17**	53.97	83.56	**80.11**	**86.62**	**78.64**
Difference	-1.45	+0.14	-0.79	+0.95	+0.30	-0.63	-0.51	+0.08	+3.02	+1.47

REFERENCES

Bosse, S., Maniry, D., Müller, K., Wiegand, T., and Samek, W. (2016). Deep neural networks for no-reference and full-reference image quality assessment. *CoRR*, abs/1612.01697.

COCO, M. (2017). Ms coco detections leaderboard.

Dai, J., Li, Y., He, K., and Sun, J. (2016). R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., and Murphy, K. (2016). Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Larson, E. and Chandler, D. M. (2009). Consumer subjective image quality database.

Li, Y., Qi, H., Dai, J., Ji, X., and Wei, Y. (2016). Fully convolutional instance-aware semantic segmentation. *CoRR*, abs/1611.07709.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). *Microsoft COCO: Common Objects in Context*, pages 740–755. Springer International Publishing, Cham.

Ponomarenko, N., Ieremeiev, O., Lukin, V., Egiazarian, K., Jin, L., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., and Kuo, C. C. J. (2013). Color image database tid2013: Peculiarities and preliminary results. In *European Workshop on Visual Information Processing (EUVIP)*, pages 106–111.

Redmon, J. and Farhadi, A. (2016). YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Schroff, F. (2009). *Semantic Image Segmentation and Web-supervised Visual Learning*. University of Oxford.

Sheikh, H. R., Sabir, M. F., and Bovik, A. C. Live image quality assessment database release 2.

Sheikh, H. R., Sabir, M. F., and Bovik, A. C. (2006). A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451.

Shrivastava, A., Gupta, A., and Girshick, R. B. (2016). Training region-based object detectors with online hard example mining. *CoRR*, abs/1604.03540.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

Tokui, S., Oono, K., Hido, S., and Clayton, J. (2015). Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., and Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171.

Zhang, C. and Ma, Y. (2012). *Ensemble Machine Learning*. Springer US.

Zhang, X., Yang, Y.-H., Han, Z., Wang, H., and Gao, C. (2013). Object class detection: A survey. *ACM Comput. Surv.*, 46(1):10:1–10:53.