Region of Interest Autoencoders with an Application to Pedestrian Detection

Jerome Williams[1], Gustavo Carneiro, David Suter
School of Computer Science
University of Adelaide
5005 Adelaide, Australia

***Abstract*— We present the Region of Interest Autoencoder (ROIAE), a combined supervised and reconstruction model for the automatic visual detection of objects. More specifically, we augment the detection loss function with a reconstruction loss that targets only foreground examples. This allows us to exploit more effectively the information available in the sparsely populated foreground training data used in common detection problems. Using this training strategy we improve the accuracy of deep learning detection models. We carry out experiments on the Caltech-USA pedestrian detection dataset and demonstrate improvements over two supervised baselines. Our first experiment extends Fast R-CNN and achieves a 4% relative improvement in test accuracy over its purely supervised baseline. Our second experiment extends Region Proposal Networks, achieving a 14% relative improvement in test accuracy.**

I. Introduction

The detection of visual objects is one of the most studied problems in computer vision [1]. A particularly relevant example of this problem is the detection of pedestrians, which is an important task in the self-driving car industry [9]. With rapidly improving hardware and increasingly large annotated datasets, we are able to apply powerful machine learning techniques to real-world detection problems. The main methodology being explored for the task of pedestrian detection is based on deep learning models [22][12][6][13], where the main challenge lies in the adaptation of such models to the unique setup of the datasets available for training.

Deep learning models use machine learning to simultaneously learn features that represent useful characteristics of the data, and a classifier to distinguish between classes. Datasets include a *training set* that the model learns from, and a *testing set* that is used to test the model's accuracy on new data. A model's accuracy on the training set measures how successful the training process was at minimizing its cost function. Accuracy on the testing set measures how well the model generalizes to new examples, and indicates how well the model will do when deployed. Improving deep learning models can be done by addressing training or generalization. Better training will improve the training accuracy, but this is only useful if the testing accuracy improves with it. Better generalization closes the gap between training and testing accuracy; this is useful as long as such a gap exists. In both cases the key measurement of success is accuracy on the testing set.

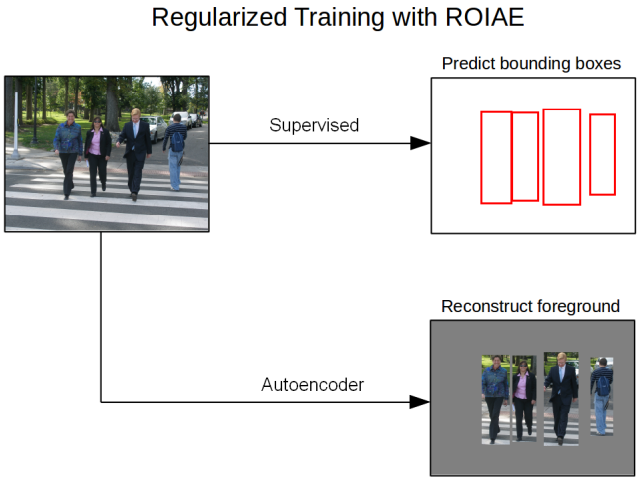

Fig. 1. The Region of Interest Autoencoder combines supervised and reconstruction tasks.

Pedestrian detection datasets are typically based on images showing a small number of pedestrians (identified with a bounding box). Positive examples come from the region inside the bounding box, and negative examples come from bounding boxes sampled from elsewhere in the image. The most common datasets used as benchmarks in pedestrian detection include Caltech-USA [7][8], ETH [30], Daimler [31] and KITTI's pedestrian data set [29].

In these datasets, the number of positive bounding boxes corresponding to pedestrians are overwhelmingly smaller than the number of candidates corresponding to background, resulting in a severe class imbalance problem. Training deep learning models usually relies on a relatively balanced set of examples in order to stop the training from collapsing to a trivial solution, such as predicting everything as background. The small number of pedestrians also hurts generalization because the few examples the model has for the positive class (i.e., pedestrians) only allows it to represent a limited amount of variations. Conversely, the presence of a large

[1] jerome.williams@adelaide.edu.au

amount of background data increases the variation in the distribution of negative samples.

The *training* difficulties of class imbalance are addressed by existing methods such as R-CNN [1], which uses sampling to present a more balanced set of samples to the detector while training. To improve *generalization* with a small amount of pedestrian samples is harder. More training samples can be gathered, the model parameters can be regularized or the problem can be re-parameterized with a more adequate model architecture. Gathering and annotating more data is effective, but costly and time consuming. This is particularly true for pedestrian detection, where data collected usually has so few positive examples. Regularizing model parameters is a widely applied solution but existing regularizers, such as weight decay, are too general to represent well the characteristics of the data set. Re-parameterization aims at improving training without increasing the number of parameters, by changing the training method and the model structure. Examples include the changes to the R-CNN model parameterization by Fast R-CNN [2] and Faster R-CNN [3], which are designed specifically for detection applications. Specializing further, for pedestrian detection instead of general detection, we have hybrid algorithms that incorporate boosted decision forests [22][12][11], and specialized part-based models that integrate prior knowledge of human structure [36]. However, increasing specialization for the pedestrian detection problem relies on increasing amounts of human effort to find good priors and heuristics, to some extent replacing the machine learning it is meant to improve. We argue that it is more interesting and useful for the computer vision community to develop a regularization approach that can be applied more generally in other detection problems. Our proposed regularization is based on a reconstruction objective function that has characteristics of all of the above methods. More specifically, we extend the training of supervised pedestrian detection models with autoencoders for image reconstruction. Weight sharing between the supervised detector and autoencoders improves the accuracy of the pedestrian detection model on the test set. The training of supervised deep learning models with autoencoders has been successful on classification tasks such as CIFAR [16] and ILSVRC [17], but to the best of our knowledge this approach has not been used to assist detection problems.

Our proposed methodology combines modern supervised deep learning detection models with an autoencoder model to form a novel deep learning approach for pedestrian detection (see Fig. 1). We call this combined model the Region of Interest Autoencoder (ROIAE). Unlike previous supervised and autoencoder combinations, we restrict our autoencoder task to only reconstruct the image's foreground regions. Because of the scarcity and smaller variation among pedestrians, this method makes the reconstruction task easier to train by efficiently using the model capacity. We demonstrate our model using the Caltech-USA dataset [7][8] for training and testing. Using the supervised detector of [6] as a baseline, our proposed method achieves a 14% log-average miss rate using a VGG-16-based Region Proposal Network (RPN) [3] (the published detection result from [6] was 14.9%). Training this baseline model with our proposed ROIAE method instead yields 12% log-average miss rate (a 2% absolute and 14% relative improvement over the purely supervised version). We also test another baseline using supervised Fast R-CNN and AlexNet that achieves a 23% log-average miss rate, while the ROIAE-trained version yields 22%. An important observation is that the autoencoder component is only used to regularize the training process. After training, the autoencoders can be discarded, leaving a detector with better accuracy on the test set and without any extra overhead in evaluation, allowing us to continue running the detector in real time (5 frames per second) on commodity hardware.

II. Literature Review

Our contribution is to take an existing pedestrian detector and to regularize its training process with autoencoders to improve generalization. In this section we summarize the literature of each area that the ROIAE draws upon: detection with deep learning and regularized training with autoencoders.

A. Detection with Deep Learning

Detection can be viewed as classification over regions in an image. A sliding window detector moves a 'window' over the input and classifies image patches into foreground or background using this window. To ensure every possible object is covered, the collection of windows needs to cover the whole image and overlap with a limited stride and with multiple scales. While this is a natural approach to solve this problem, in practice converting one input image into several thousand and classifying each one separately can be too slow, if not algorithmically optimized.

The first successful deep learning model for general detection was the Region-Based Convolutional Neural Network (R-CNN) [1], which re-parameterizes the convolutional neural network (CNN) classifiers such as AlexNet [18] and VGG-16 [20] for the problem of pedestrian detection. In R-CNN, an external method is used to propose potential regions of interest (ROIs) based on 'objectness' [1]. These ROIs are cropped out and warped to fit into the input of a neural network classifier. This is an improvement over pure sliding window because of the reduced number of background proposals, but the external proposal method itself is computationally complex. This method was extended by Fast R-CNN [2] which reuses convolutional feature maps for all ROIs in an image. The Faster R-CNN method [3] replaces the external region proposal mechanism with a deep learning based approach, called *Region Proposal Networks* (RPN). This means that the features for ROI proposal and classification can now be trained and tested in an end-to-end manner. This end-to-end

training can potentially find better candidate ROIs, because the final detection loss is used in the optimization of the RPN, so the RPN is trained to produce optimal ROI candidates. R-CNN was successfully adapted for the Caltech-USA pedestrian detection dataset by Hosang et al. in 2015 [11], achieving competitive results. In particular, Hosang et al. [11] used a boosted decision forest for region proposal, making this a hybrid model. This idea was extended by the "Scale Aware Fast R-CNN" model [12] which achieved state of the art results by creating distinct sub-networks to detect small-sized pedestrians, again using a boosted decision forest to generate region proposals. The CompACT-Deep [22] model took an alternative approach and used pretrained neural network features in a boosted decision forest detector.

Zhang et al. [6] demonstrated that it is possible to train an accurate pedestrian or other single class detector, *using region proposal networks alone*. In adapting the model to pedestrian detection they needed to address problems caused by small-sized pedestrians. While they are able to integrate R-CNN to improve the baseline RPN detector, this is only possible with *a trous* convolution to generate higher resolution outputs [6], while adding Fast R-CNN to the RPN actually degrades performance even with *a trous*. They then extended their work by creating an RPN/decision forest hybrid model, but the novelty was the ability of the RPN to get accurate pedestrian detection results on its own. Because of its speed, competitiveness and the fact that it is a pure deep learning method, we use the RPN as our main baseline model.

The RPN approach was extended by the current state of the art for pedestrian detection, the Fused Deep Neural Network (F-DNN) [13]. F-DNN uses model fusion to create a fast and accurate detector based on the Single Shot Multibox Detector (SSD) [14]. While they also provide a fast neural network only detector, F-DNN's contribution is a different supervised architecture while ours regularizes an existing supervised network with autoencoders at training time. These, along with other approaches such as decision forest hybrids, hard negative mining, ensembles and use of visual flow are not mutually exclusive and can be combined in principle.

B. Regularized Training with Autoencoders

In order to improve detection accuracy we look at previous works that explored the extension of supervised training with autoencoder learning tasks. This approach was explored for the training of the Deep Belief Network or DBN [4]. The DBN model is based on a series of smaller models called Restricted Boltzmann Machines (RBM), a type of energy-based bipartite graphical model. Training an RBM involves the minimization of a layer-wise reconstruction loss. After training one RBM, the parameters can be fixed and a second RBM is placed on top of the first and trained using the output of the hidden layer from the first RBM in a process called 'generative pre-training'. At first, generative pre-training was just used to overcome the difficulties of training deep neural networks via gradient descent, but Erhan et al. [5] demonstrated that generative pre-training can also improve generalization.

This property was exploited by the Stacked Denoising Autoencoder (SDAE) [28], which added noise to its training data and learned to reconstruct the denoised input. The SDAE learned to preserve information along vectors of variation within the training dataset, but to throw away irrelevant information. This results in a contraction of any inputs towards a manifold in feature space that contains the training data. By learning these invariants, the model generalized better than prior stacked autoencoders or the RBM-based networks.

SDAEs were initially proposed as fully-connected networks [28], but in order to exploit the potential of autoencoder-assisted learning, these results were extended to CNNs. The main problem here is that the pooling operation in CNNs is not invertible, leading to difficulty in reconstruction [25][16][33]. We base our ROIAE's autoencoder component on the Stacked What-Where Autoencoder (SWWAE) [16]. This model uses unpooling to partially invert the max-pooling process by saving the location of the pooled pixels in the original image. SWWAEs do not use generative pre-training; instead all the partial autoencoders and an additional end-to-end autoencoder are trained jointly with the supervised task. The cost function for training is a weighted sum of the cost of the supervised task and all the autoencoders. The SWWAEs showed improvement on the CIFAR [37] and STL [38] datasets when unlabelled data was used in addition to the standard labelled data. While they do perform reconstruction, SWWAEs do not use denoising or other contractive objectives to ensure contraction to a manifold. By contrast, Ladder Networks [33] use noisy skip connections in place of unpooling to recover information lost in downsampling.

Recently, SWWAE was applied to the ILSVRC classification problem, demonstrating the scalability of the model [17]. Joint training performed better than generative pre-training, and both end-to-end and intermediate autoencoders were required for best results. This model did not use additional unlabelled data; the autoencoders were able to exploit information in the labelled data that the supervised training could not. Interestingly, this model improved both training and testing accuracy, suggesting that autoencoders can improve convergence and generalization at the same time.

Our ROIAE method jointly trains an RPN for pedestrian detection with autoencoders inspired by the SWWAE described above. Unlike the methods mentioned above, we force our model to focus on the foreground (i.e., pedestrian)

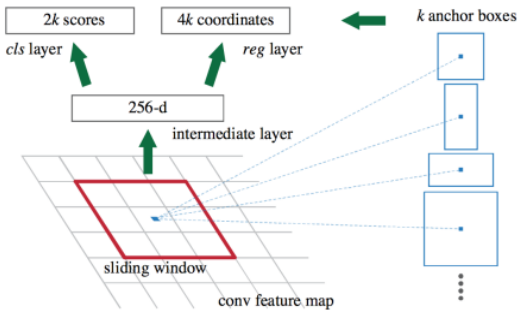

Fig. 2. Region Proposal Network, from [3]. The RPN implements a sliding window over a CNN's convolutional feature map and, for each location in the output feature map, predicts bounding boxes relative to k fixed proposals called anchors.

examples, using its limited representation capacity as efficiently as possible for our purposes. Using our proposed method we can improve detection accuracy by making up for the shortage of foreground data in detection problems.

III. Methodology

In this section, we first describe the existing supervised RPN detector in isolation. We then describe our ROIAE method, which extends the RPN during training by joint training with autoencoders that reconstruct foreground examples.

A. Detection with Region Proposal Networks

An RPN is a type of fully-convolutional network (FCN) [34], which is fine tuned from a widely available classification model such as VGG-16 [20] that has been pre-trained on the Imagenet [35] classification task. Assume that our dataset is represented by $\mathcal{D} = \{(I, \mathcal{B})_i\}_{i=1}^{|\mathcal{D}|}$, where $I : \Omega \mapsto \mathbb{R}^3$ defines an image, $\Omega \in \mathbb{R}^{H \times W}$, and $\mathcal{B} = \{b_i\}_{i=1}^{|\mathcal{B}|}, b = [x_1, y_1, x_2, y_2] \in \mathbb{R}^4$ defines a set of manually annotated bounding boxes. The RPN implements a deep learning model represented by a sequence of L pairs of linear and nonlinear transforms: $f(I, \theta) = f^L \circ f^{L-1} \circ f^{L-2} ... \circ f^1(I, \theta)$, where θ denotes the model parameters.

During testing, the model takes a test image $\tilde{I}$ as input and returns $k \times n$ bounding boxes: $\{(\tilde{b}, \tilde{c})_i\}_{i=1}^{k \times n} = f(\tilde{I}, \theta)$, where $\tilde{c} \in [0, 1]$ denotes a confidence value for each bounding box $\tilde{b}$. The output of the final layer L consists of $k \times n$ bounding boxes from the output of f^{L-1}, where n is the size of the input feature maps to layer L and k is the number of channels in L. These k channels correspond to 'anchor boxes', which are prototype bounding boxes with a preset aspect ratio and scale (see Figure 2). The predicted bounding boxes $\tilde{b}$ are defined relative to their corresponding anchor. The t-highest predicted bounding boxes are chosen as candidates, where $t < k$ is a hyper-parameter. Greedy Non-maximum suppression (NMS) is applied to prevent multiple detections for the same object, then a second, tighter confidence threshold is applied to get the final $s < t$ predictions, where s is another hyperparameter. For further details on the parameterization of the RPN see [3], [2] and [1].

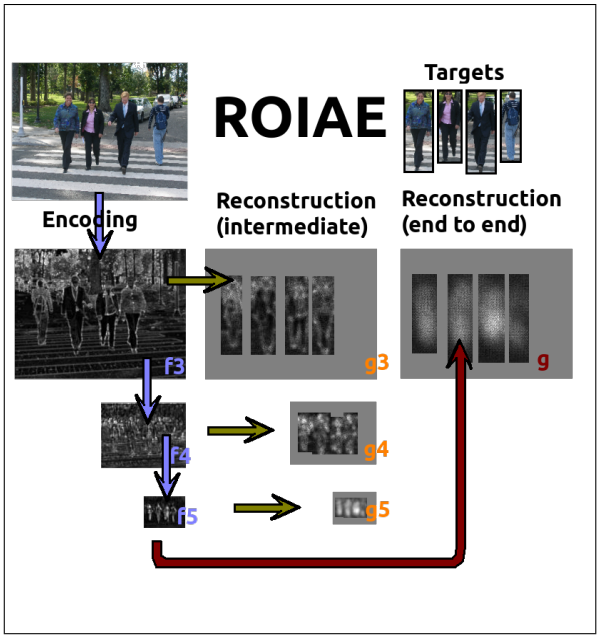

Fig. 3. Average responses of convolutional layers in an ROIAE to an annotated image. Three encoding layers are part of both a supervised RPN and the autoencoders. In this specific model we do not use g^1 or g^2. Intermediate reconstruction is generated by the autoencoders $g^3 \circ f^3, g^4 \circ f^4, g^5 \circ f^5$. The end-to-end autoencoder $g \circ f$ reconstructs f^2 rather than I.

To train the model and find θ, we use a training set extracted from dataset $\mathcal{D}$ defined earlier, containing images and ground-truth bounding boxes. There are two loss functions, a classification loss function L_{cls} and a regression loss function L_{reg}. The regression loss compares the s bounding boxes in $\tilde{\mathcal{B}}$ to the ground truth bounding boxes in $\mathcal{B}$. Each $\tilde{b} \in \tilde{\mathcal{B}}$ is assigned its nearest counterpart in $b \in \mathcal{B}$ as a regression target (see [1]).

$$L_{reg}(B, \tilde{B}) = \sum_{i=1}^{|B|} \begin{cases} ||\tilde{b}_i - b_i||_2, & \text{if } ||\tilde{b}_i - b_i||_1 \leq 1 \\ ||\tilde{b}_i - b_i||_1, & \text{otherwise} \end{cases}. \quad (1)$$

The classification loss L_{cls} addresses each $\tilde{b}$'s confidence level $\tilde{c}$. The confidence target c depends on the accuracy of $\tilde{b}$.

$$c = \begin{cases} 1, & \text{if } \frac{\tilde{b} \cap b}{\tilde{b} \cup b} > 0.5 \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

L_{cls} uses the softmax loss:

$$L_{cls}(c, \tilde{c}) = c - \frac{e^{\tilde{c}}}{\sum_{i=1}^{n} e^{\tilde{c}_i}}. \quad (3)$$

The final supervised RPN loss is a weighted sum $L_{RPN} = L_{cls} + \lambda L_{reg}$. The parameters are updated using this weighted loss with Stochastic Gradient Descent (SGD). The RPN is

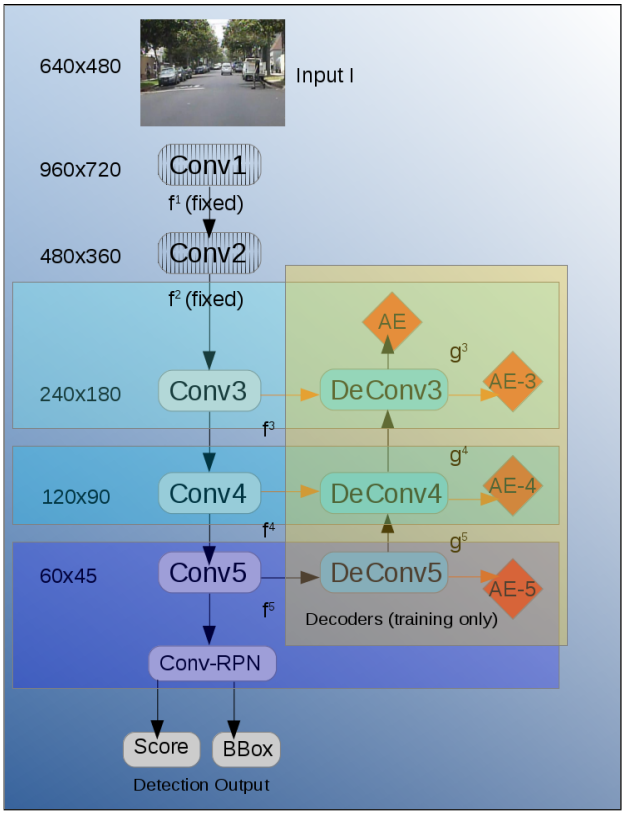

Fig. 4. VGG-16 RPN extended with Region of Interest Autoencoder. Orange arrows indicate intermediate reconstructions.

used both as our baseline and as the supervised component in our ROIAE.

B. Region of Interest Autoencoder (ROIAE)

Our ROIAE model extends the RPN by adding an autoencoder component that minimizes an image reconstruction loss. Unlike previous autoencoders we only want to reconstruct areas of an input image I within the ground truth bounding boxes $b \in B$ from our detection problem. To achieve this we construct a binary mask $M^{(I)} : \Omega \mapsto \{0, 1\}$, where 0 denotes background and 1 denotes foreground (i.e., regions containing pedestrians). The masked image is defined by $I^{(M)} = I \odot M$. In the autoencoder we use the RPN's set of transforms $f(I, \theta)$ as encoders, and we introduce L new transforms $g(I, \theta) = g^1 \circ g^2 ... \circ g^L \circ f(I, \theta)$ as decoders, $l \in \{\mathbb{N} | 1 \leq l \leq L\}$. By abuse of notation we use f^l and g^l to refer to both the function itself and the output of the function.

Once the model is trained we expect $I^{(M)} \approx I^{*(M)} = M \odot (g(I, \theta))$, where I^* is a reconstruction of I (see Figure 3 for examples of reconstructions). $g(I, \theta)$ is called the *end-to-end autoencoder*. We also define autoencoders that reconstruct encoded images f^{l-1} from a deeper encoding f^l, via decoder g^l. The encoder f^l and a decoder g^l can be turned into the *intermediate autoencoder* $g^l \circ f^l$ (if $l = 1$ the autoencoder $g^1 \circ f^1$ reconstructs I, the input to f^1). The supervised detector and autoencoder reconstruction models share f and both contribute to training its parameters (Fig. 4 depicts the proposed training structure using the VGG-16 based RPN). The loss function is

$$\begin{aligned} L_{ROIAE} = L_{RPN} \\ + \sum_{l=1}^{L} \lambda_l ||M^{(I)} \odot (f^{l-1} - g^l)||_H \\ + \lambda_{end} ||M^{(I)} \odot (I - (g^1 \circ g^2 ... \circ g^L \circ f^L))||_H, \end{aligned} \tag{4}$$

where the H is the Huber norm

$$||x||_H = \begin{cases} ||x||_2, & \text{if } ||x||_1 \leq 1 \\ ||x||_1, & \text{otherwise} \end{cases}. \tag{5}$$

The ROIAE is trained with the same scheme as the RPN, with SGD. Our autoencoder loss function uses the l_1 norm to keep the magnitude of the loss (and thus the gradient descent steps) relatively small, to ensure numerical stability. The masking operation is critical in enabling the autoencoders to discover variations that characterize the sparse foreground. During training, the decoding layers in g learn to reconstruct foreground examples and forces the encoding layers in f to update themselves to preserve information necessary for reconstruction. Preserving the variations within the foreground aims to regularize the training process. At test time the decoder layers are discarded, leaving a model of the same size as the RPN baseline but with better-regularized features.

IV. Experiments

In this section we describe the Caltech-USA pedestrian detection dataset used to evaluate our method, describe our experiments in detail and present our results.

A. Caltech-USA dataset

Caltech-USA [7][8] is one of the main benchmarks for pedestrian detection. Caltech-USA not only provides a dataset, but a detailed evaluation protocol for comparing performance. Approximately 10 hours of video have been annotated from a car driving on-road. In total there are 250,000 annotated frames, where every 3rd frame is sampled for training. Pedestrians are divided into three scales based on the height of their bounding box: near (80 or more pixels), medium (30 to 80 pixels) and far (30 pixels or smaller). The creators of the Caltech-USA dataset propose a "Reasonable" subset of the data. This set only includes pedestrians that are labeled 'person' (thus no one in crowds), at least 65% of the pedestrian visible and heights of 50 pixels or larger. This is the dataset most used as a benchmark, and we use it for all our evaluations. We treat the video frames as still images and do not make use of any temporal information.

The detector is evaluated based on the bounding boxes it returns (after non-max suppression). Bounding boxes generated by the detector must be assigned one-to-one to

TABLE I

LOG-AVERAGE MISS RATE ON CALTECH-USA

Model	Test	Train
(AlexNet)		
SCF + R-CNN (published by [11])	23.33%	N/A
ACF + Fast R-CNN	23.08%	13.38%
ACF + Fast R-CNN + ROIAE	**22.14** %	**10.28**%
(VGG-16)		
RPN (published by [6])	14.9%	N/A
RPN + R-CNN (published by [6])	13.1%	N/A
RPN (our implementation)	13.96%	12.38%
RPN (without BatchNorm)	15.42%	**7.03**%
RPN + SWWAE	13.87%	10.88%
RPN + ROIAE	**11.97**%	7.80%

the ground truth bounding boxes. Two bounding boxes can only be matched if their intersection-over-union (IOU) is 50% or higher. Ground truth bounding boxes with no match are counted as false negatives. Predicted bounding boxes with no match are counted as false positives. Ground truth bounding boxes marked 'People' (meaning a dense crowd) are set to 'ignore'. Ignored ground truth boxes can match multiple proposals at any IOU if they have not already been matched to a positive; neither the ignored ground truth or proposals matched to them are counted in the evaluation. Evaluation is presented as an f-ROC curve plotting false positives per image (FPPI) against the miss rate. Accuracy can be summarized with a scalar by taking the log-average miss rate between 10^{-2} and 10^{0}. In practice this gives similar results to the miss rate at 10^{-1} FPPI.

B. Experimental Setup

We demonstrate the ROIAE by extending two pedestrian detection models, a small scale model based on AlexNet and a larger one based on VGG-16. These models were created for the ILSVRC competition and are thus well known and pretrained versions are widely available.

1) AlexNet: Our initial work was inspired by the work of Hosang et al. [11], which used a boosted forest to provide region proposals for an R-CNN network based on AlexNet. As in [11] we use an ACF-based [26] decision forest to prepare region proposals, and a Fast R-CNN detector based on AlexNet. We use the implementation of Fast R-CNN from [2] and modify it to support the Caltech-USA dataset. Our AlexNet model is first pretrained on Imagenet [35], then on Pascal-VOC [10]. We scale up the input images to 1500x1000 with bilinear upsampling to prevent the CNN from downsampling excessively. Our AlexNet variant contains Batch Normalization (BatchNorm) modules that are not present in the original model to help with the autoencoder training. We load the features from the AlexNet variant pre-trained on PASCAL-VOC from [2] and let it adapt to the normalization during training. AlexNet has 5 convolutional layers and 2 fully connected layers. To create

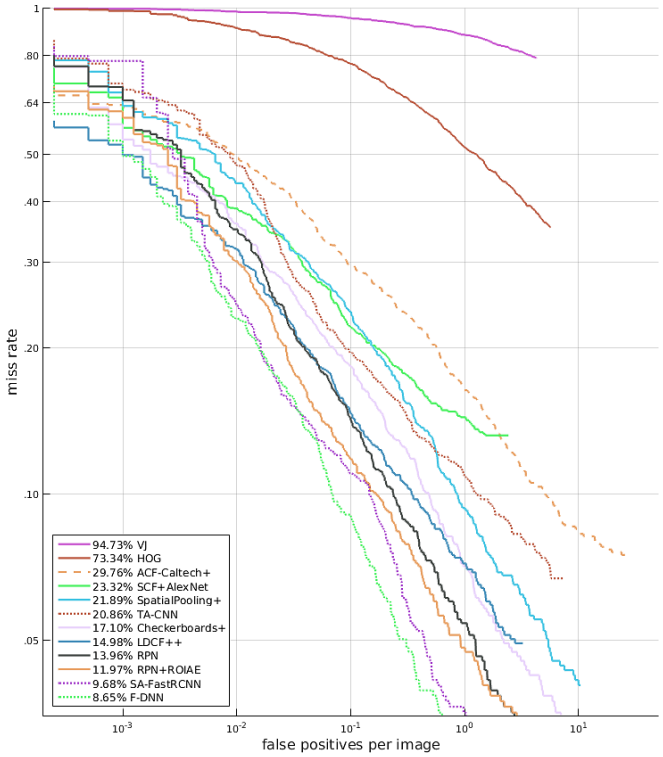

Fig. 5. f-ROC curves of our model (RPN+ROIAE) with the state of the art on Caltech-USA.

the autoencoder component of our ROIAE we construct intermediate autoencoders for convolution layers 3,4 and 5. Each intermediate autoencoder reconstructs the input of one of the convolutional layers from its output. There is also an end-to-end autoencoder that reconstructs the output of convolution layer 2 from convolution layer 5. The end-to-end and intermediate autoencoders share parameters. The end-to-end autoencoder loss has a weighting of 5×10^{-5} and the intermediate losses have a weighting of 1×10^{-5}. As in [17] the weights need to be adjusted so that they regularize the training without over-regularizing and hurting convergence on the supervised detector. We train for 40,000 iterations (here, one iteration corresponds to the training of a mini-batch) using Nesterov momentum at 0.99%. Training beyond 40,000 iterations does not improve performance. We find Nesterov momentum with a high value necessary for the relatively small AlexNet to find good search directions. We use a batch size of 4 for our training, but batch sizes from 2 to 16 did not produce any noticeable change in the results.

2) VGG-16: We replicate the setup of Zhang et al.[6] for our Region Proposal Network baseline and the supervised component in our ROIAE. Like our Fast R-CNN model, the images are scaled up, this time to 960x720, the scale used by [6]. This model is a modified version of VGG-16, containing 5 different convolutional scales, sometimes called 'macro layers'. The first 2 macro layers contain 2 sequential convolutional layers, and the third, fourth and

fifth macro layers contain 3 convolutional layers each. Each macro layer is followed by max pooling. We initialize the weights from the VGG-16 model and fix the first two macro layers in place. We insert Batch Normalization after every macro-layer. While large scale autoencoder regularization is possible without batch normalization (see [17]), it makes the model more tolerant of a range of hyper-parameters.

The autoencoder component of the ROIAE uses an end-to-end reconstruction that reconstructs the input of macro-layer 3 from the output of macro-layer 5, and has intermediate reconstruction objectives for each learnable macro-layer (i.e., macro-layers 3,4 and 5, see Figures 3 and 4). We use parametric ReLU [40] in the decoder activation functions to help train the decoder layers and apply Dropout [21] with a value of 0.5 to all data entering each decoder. We train using SGD with Nesterov momentum 0.9 using a batch size of 1 (required by the RPN implementation) for 80,000 iterations, as in [6]. We start with a learning rate of 10^{-3} and reduce it to 10^{-4} after 60,000 iterations (again following from [6]). We use a loss weight of 5×10^{-7} for the end-to-end autoencoder, 1×10^{-7} for the macro-layer 3 autoencoder, 1×10^{-9} for the macro-layer 4 autoencoder and 1×10^{-7} for the macro-layer 5 autoencoder. We found the loss weights by manual search.

We use the standard settings for Caltech, but we expand the ground truth available for training the autoencoder component by including boxes labelled 'ignore', which are usually excluded from training entirely, because these pedestrians are too close together to distinguish, too small, near the image border or too heavily occluded.

We build our RPN+ROIAE model in the Caffe framework [15] using the Matcaffe wrapped to integrate with Matlab. We use the Matlab implementation of RPN and Faster R-CNN provided by [6].

C. Results

We compare the training and testing results of our baseline and autoencoder-augmented neural network models using Dollar's Caltech toolkit in Matlab to evaluate our model [7].

The results in Table I indicate that our proposed ROIAE improves the testing and training accuracy over the purely supervised baseline. The ROIAE achieves a 1 percentage point improvement in log-average miss rate for AlexNet and a 2 percentage point improvement for VGG-16, with relative improvements of 4% and 14%, respectively. For a comparison with state of the art see the f-ROC curves in Fig. 5.

By contrast we tested a non-masked autoencoder, identical to the ROIAE except that it did not mask out the background. This non-masked autoencoder training produced nearly the same result as the baseline: 13.87% vs the baseline's 13.96% (see SWWAE in Table I). This validates the motivation behind the ROIAE: due to the small variation in the pedestrians compared to the large variation in background, the autoencoder can be trained more effectively to reconstruct the set of pedestrians only. No advantage accrues to reconstructing the background - we suppose that this happens because there is already a large amount of background samples for the supervised component to exploit. Thus reconstructing pedestrians explores the most useful reconstruction targets for detection and makes best use of the model capacity.

We performed our evaluations on an Nvidia 980 Ti GPU. The Fast R-CNN models using AlexNet take 110 milliseconds to evaluate each image (9 frames per second), while the Region Proposal Networks using VGG-16 take 175 milliseconds (5.7 frames per second) for each image.

V. Conclusion

Pedestrian Detection is one of the fastest growing applications in computer vision and machine learning. With this in mind we have not chosen to aim for record breaking results, but to demonstrate that our ROIAE yields clear improvements over an existing baseline. The ROIAE could potentially be combined with other advances in detection to yield a model of higher accuracy without any additional test-time overhead.

Regularizing training with autoencoders was an important step forward in training fully connected deep neural networks. Advances made in the training of large convolutional networks have produced networks of immense depth. However there is a limit to how much supervised training alone can achieve with limited training data while avoiding overfitting. In classification, autoencoder-assisted learning can extract more useful features out of the same data than purely supervised learning by explicitly modeling variations in the data set.

The recent success of autoencoder-augmented convolutional networks on CIFAR [16] and Imagenet [17], and the results we present in this paper with our ROIAE, imply that even more can be accomplished on detection, where foreground training examples are so sparse. Because in our method the decoder elements are thrown away after training, there is no added computational cost to during the testing procedure. The model can be used alone or combined with other advances such as sensor fusion [9], multi-scale networks [12], boosting [22] or hard negative mining [39].

In the future, we plan to expand our model to more datasets, explore the potential of the ROIAE under neural network compression schemes and to explore the nature of the features learned and whether they are similar to those in semi-supervised classification. Our experiments so far have

not used a denoising or other contractive criterion; using a ladder network might result in further improvements to generalization.

VI. Acknowledgments

The authors are members of the Australian Centre for Visual Technologies (ACVT). This work was supported by Industry Linkage Project LP130100521, "Intelligent Collision Avoidance System for Mobile Industrial Platforms". G.C. acknowledges the support by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016).

References

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik *Rich feature hierarchies for accurate object detection and semantic segmentation*, CVPR 2014.

[2] Ross Girshick *Fast R-CNN*, ICCV 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, NIPS 2015.

[4] Geoffrey E. Hinton, Simon Osindero, Yee-Whye Teh *A Fast Learning Algorithm for Deep Belief Nets*, Neural Computing 2006.

[5] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent *Why Does Unsupervised Pre-training Help Deep Learning?*, JMLR 2010.

[6] Liliang Zhang, Liang Lin, Xiaodan Liang, Kaiming He *Is Faster R-CNN Doing Well for Pedestrian Detection?*, ECCV 2016.

[7] A. Geiger, C. Wojek, B. Schiele, P. Perona. *Pedestrian Detection: A Benchmark*, CVPR 2009.

[8] A. Geiger, C. Wojek, B. Schiele, P. Perona. *Pedestrian Detection: An Evaluation of the State of the Art*, PAMI 2012.

[9] Cho H, Seo YW, Kumar BV, Rajkumar RR. *A multi-sensor fusion system for moving object detection and tracking in urban driving environments*, InRobotics and Automation (ICRA), 2014 IEEE International Conference on 2014 May 31 (pp. 1836-1843).

[10] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert, S. Lazebnik, M. Marszaek, C. Schmid, C. Russell, A. Torralba, C. Williams, J. Zhang, and A. Zisserman. *Dataset issues in object recognition*, In Towards Category-Level Object Recognition, pages 2948. Springer, 2006.

[11] Jan Hosang, Mohamed Omran, Rodrigo Benenson, Bernt Schiele *Taking a Deeper Look at Pedestrians*, CVPR 2015.

[12] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, and Shuicheng Yan *Scale-aware Fast R-CNN for Pedestrian Detection*, arXiv preprint 2015.

[13] Xianzhi Du, Mostafa El-Khamy, Jungwon Lee, Larry S. Davis *Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection*, WACV 2017.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg *SSD: Single Shot MultiBox Detector*, ECCV 2016.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, Trevor Darrell *Caffe: Convolutional Architecture for Fast Feature Embedding*, Proceedings of the 22nd ACM international conference on Multimedia, 675-678, 2014.

[16] Junbo Zhao, Michael Mathieu, Ross Goroshin, Yann LeCun *Stacked What-Where Auto-encoders*, ICLR 2016.

[17] Yuting Zhang, Kibok Lee, Honglak Lee *Augmenting Supervised Neural Networks with Unsupervised Objectives for Large-scale Image Classification*, ICML 2016.

[18] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS 2012.

[19] Sergey Ioffe and Christian Szegedy *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, ICML 2015.

[20] Karen Simonyan and Andrew Zisserman *Very Deep Convolutional Networks for Large-Scale Image Recognition*, ICLR 2015

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov *Improving neural networks by preventing co-adaptation of feature detectors*, arXiv preprint 2012.

[22] Zhaowei Cai, Mohammad Saberian, Nuno Vasconcelos *Learning Complexity-Aware Cascades for Deep Pedestrian Detection*, ICCV 2015.

[23] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, Yann LeCun *Pedestrian Detection with Unsupervised Multi-Stage Feature Learning*, CVPR 2013.

[24] Yann leCun, Leon Bottou, Yoshua Bengio, Patrick Haffner *Gradient-Based Learning Applied to Document Recognition*, Proceedings of the IEEE, 86(11):2278-2324, November 1998.

[25] Hyeonwoo Noh Seunghoon Hong Bohyung Han *Learning Deconvolution Networks for Semantic Segmentation*, ICCV 2015.

[26] Piotr Dollar, Ron Appel, Serge Belongie, and Pietro Peron *Fast Feature Pyramids for Object Detection*, PAMI 2014.

[27] Kunihiko Fukushima *Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*, Biological Cybernetics 1980.

[28] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol *Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion*, JMLR 2010.

[29] A. Geiger, P. Lenz, and R. Urtasun. *Are we ready for autonomous driving? The kitti vision benchmark suite*, In CVPR, 2012.

[30] A. Ess and B. Leibe and K. Schindler and and L. van Gool *A Mobile Vision System for Robust Multi-Person Tracking*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[31] M. Enzweiler and D. M. Gavrila *Monocular Pedestrian Detection: Survey and Experiments*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.31, no.12, pp.2179-2195, 2009.

[32] Rodrigo Benenson, Mohamed Omran, Jan Hosang, Bernt Schiele *Ten Years of Pedestrian Detection, What Have We Learned?*, ECCV 2014.

[33] Antti Rasmus, Harri Valpola, Mikko Honkala, Mathias Berglund, Tapani Raiko *Semi-Supervised Learning with Ladder Networks*, NIPS 2015.

[34] Jonathan Long, Evan Shelhamer, Trevor Darrell *Fully Convolutional Networks for Semantic Segmentation*, CVPR 2015.

[35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei *ImageNet Large Scale Visual Recognition Challenge*, IJCV 2015.

[36] Y. Tian, P. Luo, X. Wang, and X. Tang *Deep Learning Strong Parts for Pedestrian Detection*, ICCV 2015.

[37] Alex Krizhevsky *Learning Multiple Layers of Features from Tiny Images*, masters thesis, University of Toronto 2009.

[38] Adam Coates, Honglak Lee, Andrew Y. Ng *An Analysis of Single Layer Networks in Unsupervised Feature Learning*, AISTATS 2011.

[39] Abhinav Shrivastava, Abhinav Gupta, Ross Girshick *Training Region-based Object Detectors with Online Hard Example Mining*, CVPR 2016.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, ICCV 2015.