Application Specific Performance Measure Optimization Using Deep Learning

by

Md Atiqur Rahman

A thesis submitted to The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg

Thesis Supervisor | Author
Dr. Yang Wang | **Md Atiqur Rahman**

Application Specific Performance Measure Optimization Using Deep Learning

Abstract

In this thesis, we address the action retrieval and the object category segmentation problems by directly optimizing application specific performance measures using deep learning. Most deep learning methods are designed to optimize simple loss functions (e.g., cross-entropy or hamming loss). These loss functions are suitable for applications where the performance of the application is measured by overall accuracy. But for many applications, the overall accuracy is not an appropriate performance measure. For example, applications like action retrieval often use the area under the Receiver Operating Characteristic curve (ROC curve) to measure the performance of a retrieval algorithm. Likewise, in object category segmentation from images, the intersection-over-union (IoU) is the standard performance measure. In this thesis, we propose approaches to directly optimize these complex performance measures in deep learning framework.

Contents

List of Figures

List of Tables

List of My Publications

[1] Rahman, M. A., and Wang, Y. Learning neural networks with ranking-based losses for action retrieval. In *The 13th Conference on Computer and Robot Vision (CRV)* (June 2016).

Acknowledgments

I would like to take this unique opportunity to express my heartfelt thanks and sincere gratitude to my supervisor Dr. Yang Wang for his continuous support and guidance throughout my Master's program. Every solution presented in this thesis is directly influenced by his thoughtful insights into the problems. It was a great honor to work under his supervision, I must say. I am also indebted to Dr. Neil Bruce for being ever so kind to offer valuable advice regarding my research. My gratitude also extends to the fellow lab members of the Computer Vision lab for their generous support and genuine kindness that helped sustain a positive atmosphere in the lab. I gracefully acknowledge the financial support that I received from the Department of Computer Science and the University of Manitoba Graduate Fellowship (UMGF), as well as the Mitacs-Accelerate Graduate Research Internship Program. Finally, I extend my heartiest gratitude to the most important people in my life – my parents and my family members. Without their constant support, encouragement, and understanding, it would not have been possible for me to achieve my academic goals.

This thesis is dedicated to my parents.

Chapter 1

Introduction

1.1 General Introduction

Deep Neural Networks (DNNs) is a class of neural networks that, unlike the classical neural networks, includes many hidden layers. The lower layers in a DNN usually extract generic low-level features of the input building upon which the upper layers produce progressively more abstract representations of the input. As a result, DNNs have the ability to automatically learn high-level features which are highly representative of the input and have been reported to produce superior results compared to the conventional machine learning algorithms that rely on traditional hand-crafted features. With the improvements of Graphics Processing Unit (GPU) hardware and the availability of massive training datasets, interest in DNNs has recently been rekindled by Krizhevsky et al. [20], as they achieved huge gain over conventional machine learning methods by using a kind of DNN known as Convolutional Neural Networks (CNN) [21] for image classification in the ImageNet challenge [7].

In order to achieve good performance for any specific application, ideally we would like the learning algorithm to optimize the actual target performance measure used in that application. However, most deep networks are trained to optimize simple loss functions like the hinge loss, softmax loss or the categorical cross-entropy loss. These loss functions optimize for the overall error rate and are mostly suitable for standard classification problems, where the performance of the learning algorithm is measured by the overall accuracy. But for many applications, where the overall accuracy is not an appropriate performance measure, using such simple loss functions often results in learning the parameters against a wrong performance measure and not the actual target performance measure. As a result, sub-optimal results are produced. Therefore, directly optimizing the appropriate target performance measure is very important for the overall success of a learning algorithm.

Optimizing application specific loss functions has been studied in learning linear models so far. E.g., Joachims [19] developed methods for optimizing several application specific complex loss functions based on structural Support Vector Machine (SVM) formulation. Ranjbar et al. [27] used Markov Random Field (MRF) models to directly optimize some complex non-decomposable loss functions. But these methods are limited to linear models, and therefore, cannot capture the higher-order nonlinearities usually inherent in the data. In contrast, deep learning methods are better suited to capture the nonlinear relationship existing in the data and are reported to produce much better results than the earlier approaches.

In this thesis, we propose to directly optimize some application specific loss functions using DNNs. We specifically apply the proposed approach to two problems, human action retrieval from images/videos and object category segmentation from images.

1.2 Action Retrieval

In an action retrieval setting, given a query action of interest (such as, “walking”, “running” etc.), we would like to be able to retrieve all images/videos from a large image/video repository that are relevant to the query. The input to an action retrieval system is a set of images/videos and a query action, and the desired output is a ranked list of the images/videos according to their relevance to the query. Therefore, the action retrieval problem can be translated to assigning either “relevant”, or “irrelevant” label to each image/video in the repository depending on their relevance to the query. For such an application, learning algorithms that optimize for overall accuracy or error rate may end up learning to assign all images/videos the “irrelevant” label, as most of the images/videos in a large repository would be irrelevant to the query.

The large imbalance between the two classes (“relevant” and “irrelevant”) in the action retrieval setting can be handled by a performance measure called Receiver Operating Characteristic area (ROC area) which measures the area under the Receiver Operating Characteristic curve (ROC curve). The ROC curve is a plot of the true positive rate against the false positive rate for all possible classification thresholds. Therefore, ROC area specifies the probability that a classification decision function will rank a “relevant” example higher than an “irrelevant” example, when selected at random. Unlike overall accuracy, ROC area is insensitive to the class imbalance issue usually present in the action retrieval setting. In this thesis, we address the action retrieval problem by directly optimizing the ROC area measure using deep networks. To this end, we formulate the problem as a multivariate structured prediction problem and incorporate the ROC area loss into the learning objective of the deep network.

1.3 Object Category Segmentation

The object category segmentation problem can be defined as the task of labeling the pixels of a given image as being part of a given object (foreground) or not (background). In such a problem setting, the two classes (foreground and background) are often very imbalanced, as the majority of the pixels in an image usually belong to the background. Learning algorithms that are designed to optimize for overall accuracy may not be suitable in this problem setting, as they might end up predicting every pixel to be background in the worst case. For example, if 90% of the pixels belong to the background, a naive algorithm can achieve 90% overall classification accuracy simply by labeling every pixel as the background.

The standard performance measure that is commonly used for the object category segmentation problem is called intersection-over-union (IoU). Given an image, the IoU measure gives the similarity between the predicted region and the ground-truth region for an object present in the image, and is defined as the size of the intersection divided by the union of the two regions. The IoU measure can take into account of the class imbalance issue usually present in such a problem setting. For example, if a naive algorithm predicts every pixel of an image to be background, the IoU measure can effectively penalize for that, as the intersection between the predicted and ground-truth regions would be zero, thereby producing an IoU count of zero.

Most deep learning based methods address the image segmentation problem using simple loss functions, such as, softmax loss optimizing for overall accuracy. Therefore, they are subject to the problem mentioned above. We argue that directly optimizing the IoU loss is superior to the methods optimizing for simple loss functions. In this thesis, we address the object category segmentation problem by directly optimizing the IoU measure in a deep learning framework. And to do so, we incorporate the IoU loss in the learning objective of the deep network.

1.4 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 discusses several works related to optimizing different application specific performance measures including ROC area and IoU. It also discusses the recent advancements in deep learning for addressing the image/video retrieval and object category segmentation problems. Chapter 3 focuses on the application of action retrieval, where we propose a deep learning based approach for optimization of the ROC score. Chapter 4 focuses on object category segmentation, where we develop deep learning based approach for optimizing the IoU score. Finally, chapter 5 concludes the dissertation and mentions possible future research directions.

Chapter 2

Related Work

2.1 Application Specific Performance Measure Optimization

Methods for optimizing application specific performance measures are mostly based on the large margin structural Support Vector Machine (SVM) formulation [35] originally designed for solving structured output prediction problems. For example, Joachims [19] proposed methods for optimizing a range of nonlinear performance measures including ROC area that are a function of false positive and false negative counts.

Following the above approach, Yue et al. proposed a method called AP-SVM [38] that learns binary ranking functions by directly optimizing a straightforward relaxation of the performance measure called Average Precision (AP). AP is a performance measure which is defined as the average of the precisions taken at every correct retrieval in a ranked retrieval system. However, AP-SVM is limited to fully-supervised setting only, and not suitable for weakly-supervised data. In a weekly supervised setting, only a few of the training examples are fully labeled while others usually have some weak labeling. To adapt for weakly supervised data, Behl et al. proposed a novel latent AP-SVM formulation by optimizing a tighter upper bound on the AP loss with the help of additional annotations available as latent variables. While all of these approaches are based on linear model, a very recent work by Song et al. [31] proposed a deep learning based approach for directly optimizing the AP measure. They extended the theorem of McAllester et al. [14] to handle nonlinear models in order to compute the gradient of complex non-differentiable loss functions including AP loss, and trained a deep CNN end-to-end to directly optimize the AP measure.

Several approaches were proposed in the literature for direct optimization of ROC area, a standard performance measure for information retrieval. These works are similar to the proposed approach for action retrieval as they also use gradient descent for directly optimizing ROC area. An early work by Herschtal et al. called RankOpt [15] uses ROC area as its objective function which is then optimized using gradient descent. However, since ROC area is non-differentiable, it optimizes an approximation to ROC area based on a sigmoid function. Just like the proposed approach, Mcfee and Lanckriet [23] also based their work on structural SVM framework. They used gradient descent for metric learning interpreted as an information retrieval problem. With this setting, they provided algorithms for optimizing a set of ranking-based performance measures including ROC area.

Regarding direct optimization of the IoU measure, the first work to address this problem in computer vision was proposed by Blaschko et al. [5] with an application to object detection and localization. Based on a structured output regression model, they used joint-kernel map and proposed a constraint generation technique to efficiently solve the optimization problem of structural SVM framework. Ranjbar et al. [27] used structured Markov Random Field (MRF) to directly optimize IoU by replacing the non-decomposable loss function with a piecewise linear approximation and applied the approach for object category segmentation. They solved the loss-augmented inference problem of the structural SVM formulation by first converting it to a relaxed linear program and then solving it for each piece the loss function is segmented into. Instead of using piecewise linear approximation of the loss function, [34] addressed the problem of optimizing IoU using highly efficient special-purpose message passing algorithms.

Apart from the above approaches for empirical risk minimization for IoU, there have been some recent works based on the Bayesian decision theory that give a closed form statistical approximation to the IoU measure. For example, Nowozin [25] used a Conditional Random Field (CRF) distribution model and proposed some heuristics including a greedy heuristic to maximize the value of Expected-Intersection-over-Expected-Union (EIoEU), which is a closed form approximation to Expected-Intersection-over-Union (EIoU). Premachandran et al. [26], on the other hand, optimizes exact Expected-IoU, but over a small set of high-quality candidate solutions by approximating the joint distribution of input and output under a delta distribution. The latest work by Ahmed et al. [2] draws the best from both of these approaches. Based on the fact that the EIoEU is exact for a delta distribution, they take the idea of approximating EIoU from [26] by taking the average of EIoEU as computed in [25], but over multiple delta distributions with individual delta functions. The proposed approach for addressing the semantic segmentation problem by directly optimizing IoU follows the first of these two methods.

2.2 Image Retrieval using Deep Learning

As the image retrieval problem suffers from large semantic gap between high-level representation of textual queries and low-level representation of images, Bai et al. learned high-level representations of images by using a multi-tasking transfer learning DNN architecture [3] and trained a set of binary classifiers for different textual queries based on these representations. Since it is very difficult for such an approach to scale up with a massive number of queries, a bag-of-words (BoWs) based DNN model was proposed in [4]. Here, the high-level representations of input images learned from the DNN are mapped into BoWs space where visual similarity between images is computed. The relevance between a textual query and an image is measured by the cosine similarity between BoW representations of the two. To further improve the results, a page rank algorithm was used to consider the visual similarity of the retrieved images.

The work of Razavian et al. [28] to address the image retrieval problem is similar to our proposed approach for action retrieval, as it also exploits image representations obtained from a classification CNN. Their method does not require fine-tuning the classification CNN with target domain data, still can deliver high retrieval accuracy when compared to techniques not based on CNN image representations. A very recent work [24] in this direction of exploiting classification CNN for image retrieval showed that image representations obtained from the lower layers of the classification CNN performs better than that obtained from the last layer. Based on the recent successful classification CNNs like GoogLeNet [33] and OxfordNet [30], they leverage the benefit of using vectors of locally aggregated descriptors (VLAD) encoding of the local convolutional features obtained from the lower layers of these classification nets for instance level image retrieval.

2.3 Semantic Segmentation using Deep Learning

The semantic segmentation problem is similar to object category segmentation, but requires labeling each pixel of an image as being part of one of several semantic object categories (e.g., cow, bus, chair etc.) instead of simply foreground or background. Recently, several approaches have been proposed for semantic segmentation that take advantage of high-level representation of images obtained from DNNs. Hariharan et al. [12] used a CNN architecture that can simultaneously perform object detection and semantic segmentation which they coined as SDS (Simultaneous Detection and Segmentation). Based on some initial region proposals, they prune out the negative bounding boxes using the CNN features extracted from both the bounding boxes of the regions as well as region foregrounds. Long et al. [22] has proposed a fully convolutional model for semantic segmentation that achieved the state-of-the-art segmentation performance. Starting with pre-trained classification CNNs (e.g., AlexNet [20], GoogleNet [33]), they replaced the fully connected layers of the CNNs with convolution layers and fine-tuned the resulting networks end-to-end with target domain data. To further boost the performance, they also proposed some skip architectures that combine coarse semantic information obtained from the last layer with fine appearance information obtained from earlier layers. Our proposed approach for object category segmentation is closely related to this approach, as it also fine-tunes a classification CNN with target domain data.

Following the above approach of coalescing coarse level semantic information with fine grained local information, Hariharan et al. [13] addressed the task of semantic segmentation by using a pixel descriptor called hypercolumn, which combines the activation for that pixel obtained from the last layer as well as the earlier layers of a CNN. They showed significant performance gain as a result of using such descriptors. Very recently, Sharma et al. [29] has proposed a novel DNN architecture for semantic segmentation that combines a CNN with a Recursive Neural Network (RNN). While the CNN aggregates bottom-up local visual features of the image and maps them into a global image representation, the RNN propagates the aggregated information top-down so that contextual information is disseminated to every spatial region of the image. As a result of this contextual information propagation, it achieves high segmentation results on some benchmark datasets.

Chapter 3

Optimizing ROC area for Action Retrieval

In this chapter, we describe the proposed approach to learn binary ranking functions that can directly optimize the ROC area measure using deep neural network with a view to addressing the action retrieval problem. Since ROC area is a nonlinear performance measure that cannot be decomposed over individual instances of a training set, we use multivariate structured SVM formulation to predict the ranking of the whole training set instead of individual instances, as described in [19]. Unlike the SVM approach, we use a deep neural network learning complex nonlinear ranking functions (Fig. 3.1).

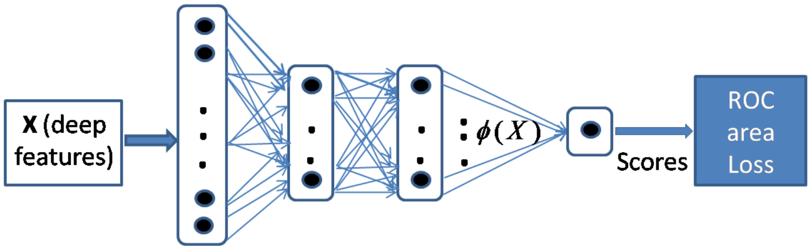


Figure 3.1: Architecture of the deep neural network for the proposed action retrieval approach.

3.1 Methodology

More formally, let $S = ((x_1, y_1), \dots, (x_n, y_n))$ represent a training set of n examples, where $x_i \in \mathbb{R}^d$ represents the feature vector for a single example and $y_i \in \{-1, +1\}$ represents one of two possible ranks of the example, namely, irrelevant or relevant. Instead of predicting the rank of each example individually, the proposed approach tries to learn a mapping function $h : X \times \dots \times X \to Y$ that takes all n examples $X = (x_1, \dots, x_n)$ at once and maps them to a vector of n labels $y = \{y_1, \dots, y_n\} \in Y = \{-1, +1\}^n$. In order to obtain the best label vector y that gives the optimal ordering of the whole training set resulting in the best ROC area measure, we define a nonlinear discriminant function p as follows:

$$p(X) = \arg\max_{y \in Y} F_W(X, y) \tag{3.1}$$

Here, $F_W(X, y)$ is a scoring function which in turn is defined as follows:

$$F_W(X, y) = W_M^T \Psi(\phi(X), y) \tag{3.2}$$

Here, $\phi(X)$ is a transformation function that performs a sequence of nonlinear transformations on the training set X. To be specific, each example $x_i \in X$ is passed through $m = 1, 2, \dots, (M - 1)$ layers of nonlinear transformations in a deep neural network, where the output of the m^{th} layer is given by -

$$v_i^{m+1} = s(W_m^T v_i^m + b_m) \tag{3.3}$$

with the initial case of $v_i^1 = x_i$. Here, W_m and b_m are the set of weights and biases, respectively at layer m and $s : \mathbb{R} \mapsto \mathbb{R}$ is a nonlinear activation function, which in our case is the *sigmoid* function. Therefore, the whole training set $X = (x_1, \ldots, x_n)$ is transformed to a nonlinear representation V^M, such that $V^M = \phi(X) = (v_1^M, \ldots, v_n^M)$.

Now, referring back to Eq. 3.2, $\Psi(\phi(X), y)$ is a compatibility function that measures the compatibility between the transformed input V^M and the output label vector y. Following [19], we also used a simple compatibility function Ψ of the following form that depends only on individual transformed training example v_i^M and its rank label y_i.

$$\Psi(\phi(X), y) = \Psi(V^M, y) = \sum_{i=1}^{n} v_i^M y_i = V^M y \tag{3.4}$$

Finally, the $(M-1)$ nonlinear layers of the neural network are followed by a linear scoring layer (the M^{th} layer) with weights W_M (and no biases) to produce the scores $F_W(X, y)$ as shown in Eq. 3.2. Therefore, putting everything together, the optimal labeling sequence for the training set X would be –

$$p(X) = \arg\max_{y \in Y} W_M^T V^M y \tag{3.5}$$

Once the scores of all the examples in the whole training set are predicted, they are sorted in descending order to get a total ranking of the whole training set. A perfect ranking requires the scores for all relevant examples to be higher than that of the irrelevant ones. In order to learn the retrieval function that minimizes the

ROC area loss of the whole training set, the deep neural network tries to optimize an objective function of the following form:

$$\begin{aligned}\arg\min_{W_m,b_m} O = &\ O_1 + O_2 \\ = &\ F_W(X,y') + \Delta(y,y') - F_W(X,y) + \frac{\lambda}{2}(\sum_{m=1}^{M} ||W_m||_F^2 + \sum_{m=1}^{M-1} ||b_m||_2^2) \\ = &\ W_M^T V^M y' + \Delta(y,y') - W_M^T V^M y + \frac{\lambda}{2}(\sum_{m=1}^{M} ||W_m||_F^2 + \sum_{m=1}^{M-1} ||b_m||_2^2)\end{aligned} \tag{3.6}$$

The objective function O includes two terms – the loss term O_1 and the regularization term O_2. Minimizing O_1 actually leads to maximizing $F_W(X,y)$ (the score for the correct label vector y) and minimizing $F_W(X,y')$ (the score for any incorrect label vector y'). Instead of an example-based loss, O_1 is having a sample-based loss $\Delta(y,y')$, which is actually an application specific loss and measures the ROC area loss in this case. The regularization term O_2 tries to keep the parameters of the neural network small. Here, $||A||_F$ represents the Frobenius norm of the matrix A and λ is a regularization parameter. Like [19], we also adopt pair-wise ranking to learn the retrieval function. Therefore, the ROC area loss in this setting can be simply measured by the number of misranked pairs as proposed in [19] and defined by the following equation:

$$\Delta(y,y') = \frac{\text{total misranked pairs}}{P \times N} \tag{3.7}$$

where, P and N represent the total number of relevant and irrelevant examples in the training set, respectively. To calculate the total misranked pairs for the current parameters, we use Algorithm 3 as described in [19].

In order to obtain the set of weights W_m (for $m = 1, 2, \ldots, M$) and biases b_m

(for $m = 1, 2, \ldots, M-1$), Eq. 3.6 is solved using stochastic gradient descent. The gradient G_M^W of the objective function O with respect to the weights of the M^{th} layer (i.e; W_M) can then be written as follows:

$$\begin{aligned} G_M^W = \frac{\partial O}{\partial W_M} &= \Psi(\phi(X), y') - \Psi(\phi(X), y) + \lambda W_M \\ &= V^M y' - V^M y + \lambda W_M \end{aligned} \tag{3.8}$$

For the other layers of the neural network, i.e.; for $m = (M-1), \ldots, 1$, the gradients G_m^W and G_m^b with respect to the weights W_m and biases b_m, respectively can be computed using the chain rule of derivatives as follows:

$$G_m^W = \frac{\partial O}{\partial V_M}\frac{\partial V_M}{\partial V_{M-1}} \cdots \frac{\partial V_{m+1}}{\partial W_m} = \delta_m V_m + \lambda W_m \tag{3.9}$$

$$G_m^b = \delta_m + \lambda b_m \tag{3.10}$$

where, δ_m is defined as follows:

$$\delta_m = \begin{cases} W_{m+1}^T (y' - y) \odot s'(Z_m) & \text{, if } m = M-1 \\ W_{m+1}^T \delta_{m+1} \odot s'(Z_m) & \text{, otherwise} \end{cases} \tag{3.11}$$

Here, $s'(a)$ is the derivative of the *sigmoid* function $s(a) = \frac{1}{1+e^{-a}}$ and $\odot$ is an element-wise multiplication operator. Z_m is defined as –

$$Z_m = W_m^T V_m + b_m \tag{3.12}$$

The update rule for the I^{th} iteration of weight update then becomes $W_m^I = W_m^{(I-1)} + \eta G_m^W$, where η is the learning rate. Update rule for the biases are similar. Details about setting the hyper-parameters η and λ are discussed in section 3.2.

3.2 Experiments

3.2.1 Setup and Datasets

In order to predict a binary ranking with a view to retrieving images or videos from a large repository, we directly optimize the ROC area measure using a deep neural network. The proposed approach is compared with an structural SVM formulation called $\text{SVM}_{\text{multi}}$ [19] (as implemented in $\text{SVM}^{\text{light}}$ [18]) that can also directly optimize ROC area. Moreover, to support the hypothesis that directly optimizing application specific loss (in this case, ROC area loss) gives better performance than optimizing some surrogate loss, we compare the proposed approach with a standard neural network having the same architecture and parameters, but optimized for general classification loss, more specifically, softmax loss. For the rest of the chapter, the proposed neural network approach directly optimizing for ROC area loss is referred to as NN_{ROC}, while the baseline approach of using a classical neural network with general classification loss (i.e.; softmax loss) is referred to as NN_{Gen}.

To evaluate the proposed approach, we conducted experiments on two different datasets – the Stanford 40 actions dataset [37] and the UCF101 actions dataset [32]. The Stanford 40 actions dataset contains 4,000 training images and 5,532 test images covering a total of 40 different human action categories. The UCF101, on the other hand, is a video dataset containing videos of 101 action classes with a train and test split of 9,537 and 3,783 videos respectively, summing up to a total of 13,320 videos. For all the experiments, the train/test splits as suggested by the datasets were used.

The deep neural networks as shown in Fig. 3.1 was used for both the proposed approach of $\mathrm{NN}_{\mathrm{ROC}}$ and the baseline approach of $\mathrm{NN}_{\mathrm{Gen}}$. The DNN consists of four layers (i.e., $M = 4$) with 100, 50, 50 and 1 units in the layers, respectively, where these parameters were chosen empirically based on a validation set. Momentum and weight decay with standard parameter settings of 0.9 and 0.0005, respectively were employed during training. We used fixed learning rates of 10^{-3} and 10^{-4} for the Stanford 40 and UCF101 datasets, respectively, as selected by line search. Because batch gradient descent is slower as it performs gradient update on the whole training set, and because stochastic gradient descent fluctuates a lot as it performs gradient update on each example, we used stochastic gradient descent in mini batches (batch size = 100) which draws the best of the two approaches. All the weights and biases of the network were initialized randomly. Both $\mathrm{NN}_{\mathrm{ROC}}$ and $\mathrm{NN}_{\mathrm{Gen}}$ were implemented using a popular deep learning tool called MatConvNet [36].

For the image dataset, we extracted 4,096 dimensional feature vector for each image from the fc6 fully connected layer of the Caffe implementation [17] of AlexNet deep network model as described in [20]. The reason for using activations from fc6 layer as feature vectors for the input images is because they have been reported to produce better results for a variety of visual recognition tasks [8].

For the UCF101 dataset, we used a deep-learning based video representation tool called Convolutional 3D (C3D) [9]. C3D is a deep 3-D convolutional network that is trained on a large scale of video dataset. It has been reported to provide state-of-the-art video representation used for video analysis. C3D segments a video into chunks of 20 frames. It then passes each chunk of frames through the deep network and extracts 4,096 dimensional deep-learning feature vector from the fully connected

Table 3.1: Retrieval performance comparison of the proposed approach with the baselines.

Dataset	Average ROC area (%)			Improvement over the baselines (%)		# of classes showing performance gain		# of classes showing performance loss	
	NN$_{\text{Gen}}$	SVM$_{\text{multi}}$	NN$_{\text{ROC}}$	NN$_{\text{Gen}}$	SVM$_{\text{multi}}$	NN$_{\text{Gen}}$	SVM$_{\text{multi}}$	NN$_{\text{Gen}}$	SVM$_{\text{multi}}$
Stanford 40 actions Train/Test: 4000/5532 Features: 4096 Total Class: 40	84.65	88.00	91.11	6.46	3.11	all	35	none	5
UCF101 actions Train/Test: 9537/3723 Features: 4096 Total Class: 101	95.12	98.66	99.13	4.01	0.47	all	51	none	34

layer fc7. Finally, the individual group feature vectors are averaged over each video to produce a single 4,096 dimensional vector representation of the video.

The regularization parameters λ (for NN$_{\text{ROC}}$ and NN$_{\text{Gen}}$) and C (for SVM$_{\text{multi}}$) were set empirically by using a validation set consisting of $\frac{1}{3}$ of the training examples selected at random. For the baseline SVM approach of SVM$_{\text{multi}}$ both linear and nonlinear kernels were used and the best results are reported.

3.2.2 Results

For each of the classes in a dataset, the proposed approach learns a binary retrieval function considering the examples belonging to the query action class as relevant and all other examples as irrelevant. Table 3.1 lists the retrieval performance on the two datasets as achieved by the proposed approach NN$_{\text{ROC}}$ and the two baseline methods of SVM$_{\text{multi}}$ and NN$_{\text{Gen}}$. For each dataset, we report the ROC area averaged across all the classes.

While comparing the approach NN$_{\text{ROC}}$ to the baseline approach SVM$_{\text{multi}}$, among the 40 classes in the image collection, 35 classes showed performance gain as opposed

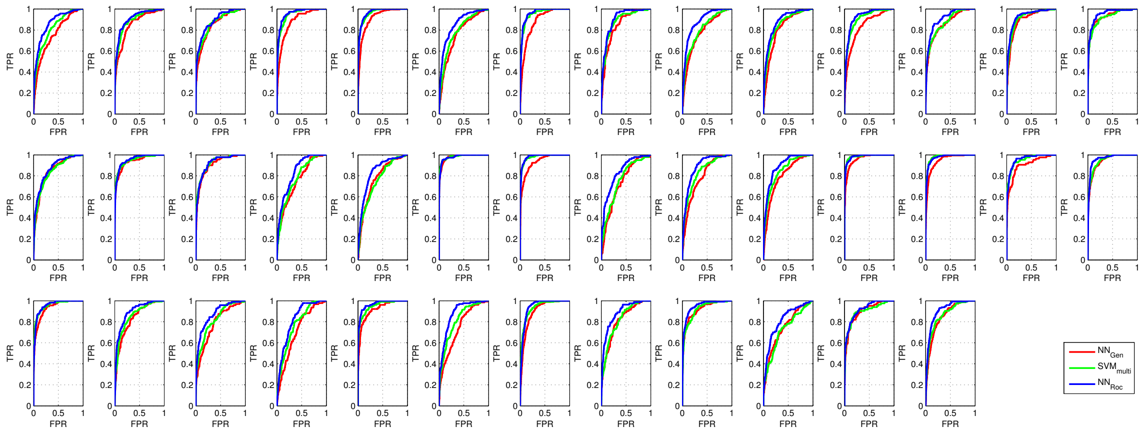


Figure 3.2: ROC curves (from top to bottom and left to right) of the proposed approach and the baseline methods for the 40 different action classes in the Stanford 40 actions dataset [37] in increasing order of class id . TPR and FPR represent True Positive Rate and False Positive Rate, respectively.

to 5 showing decline in performance, whereas, for the video collection, 51 showed performance gain, 34 showed decline in performance and the rest were affected by neither approach. On the other hand, while comparing to the other baseline approach of $\mathrm{NN}_{\mathrm{Gen}}$, all the classes for both datasets see performance improvement with an average performance gain larger than that achieved over $\mathrm{SVM}_{\mathrm{multi}}$. This is no surprise as $\mathrm{NN}_{\mathrm{Gen}}$ is not optimizing ROC area loss, rather general classification loss, and therefore, exhibits poor retrieval performance.

Figure 3.2 shows ROC curves of the proposed approach as well as the two baseline methods for 40 different query action classes on the Stanford 40 actions dataset [37]. As depicted in the figure, the ROC curves of the proposed approach (blue curves) are covering the respective ROC curves of the baseline methods for almost all classes.

Since learning binary retrieval functions requires predicting binary ranking that maximizes the scores for the relevant examples while minimizing scores for the ir-

Table 3.2: Comparison of multi-class classification accuracy of the proposed approach with the baselines.

Dataset Method	Stanford 40	UCF101
NN_{Gen}	28.53%	64.72%
$\text{SVM}_{\text{multi}}$	36.75%	70.10%
NN_{ROC}	40.62%	75.06%

relevant ones, we can perform classification by these scores. Therefore, to further investigate the effectiveness of the approach, we also perform multi-class classification by taking the scores of an example for all the classes and then predicting the class of the example to be the one that gives the maximum score. The results are shown in Table 3.2.

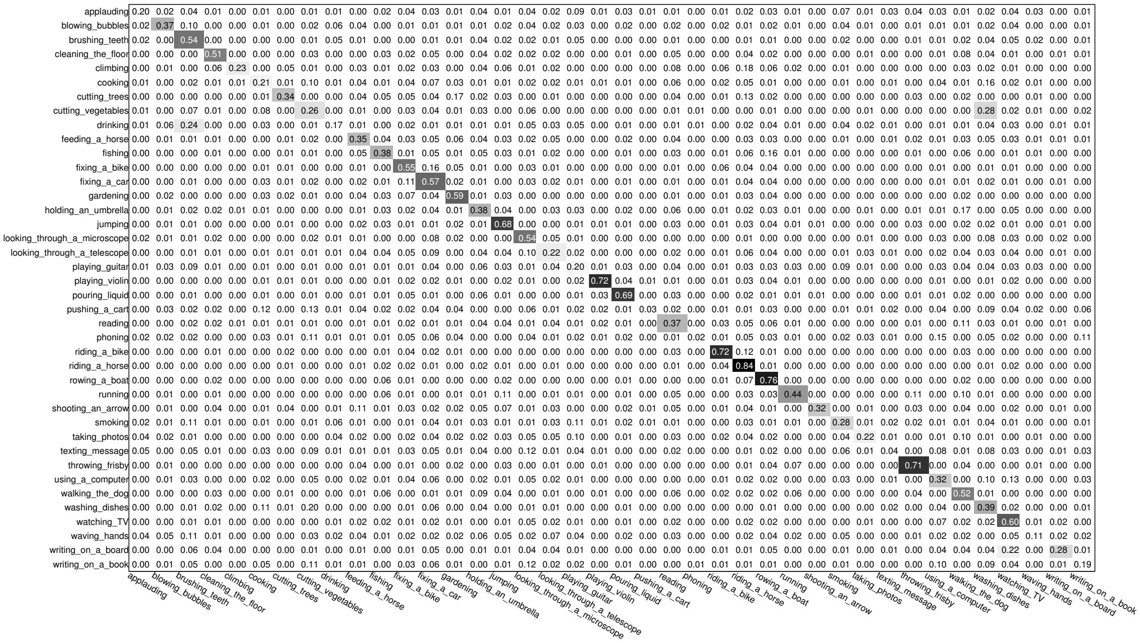
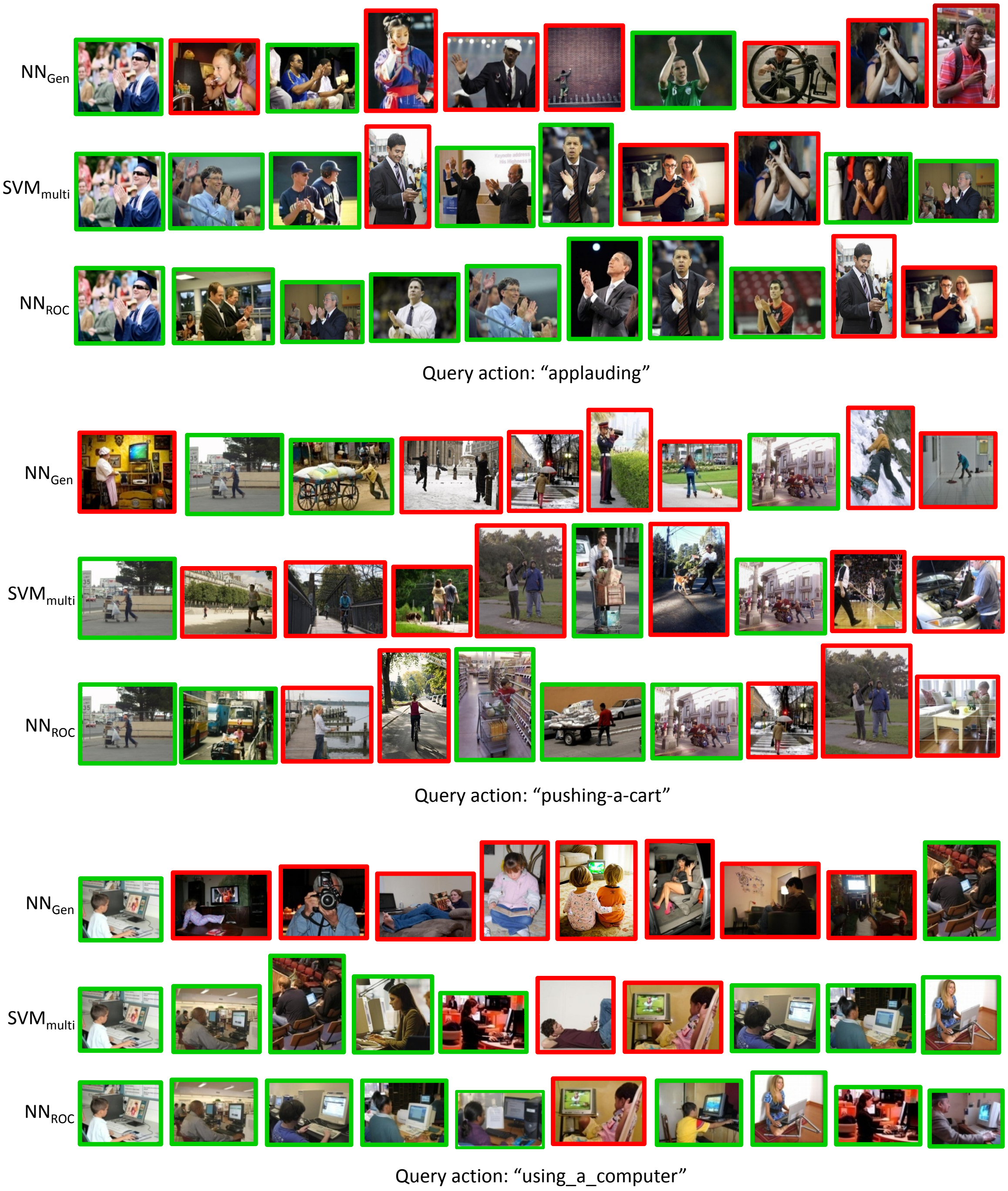


Figure 3.3: Confusion matrix of the proposed approach on the Stanford 40 actions dataset [37].

As shown in the table, the proposed approach outperforms the two baseline methods for both datasets. The improvement is more pronounced over the baseline approach of $\mathrm{NN}_{\mathrm{Gen}}$ than $\mathrm{SVM}_{\mathrm{multi}}$. This is attributed to the fact that $\mathrm{NN}_{\mathrm{Gen}}$ is optimizing for softmax loss which is a general classification loss, whereas, $\mathrm{SVM}_{\mathrm{multi}}$ and the proposed approach both optimize for application specific loss, namely ROC area loss. The reason the proposed approach demonstrates superiority over the SVM based approach is because, it provides a nonlinear model which is able to better handle the higher order nonlinearities inherent in the data. Figure 3.3 shows the confusion of the approach for the multi-class classification on the Stanford 40 actions dataset [37].

As a reference to the qualitative results produced by the proposed approach, some retrieval examples of the different methods for three different query action classes on the Stanford 40 actions [37] dataset are shown in Fig. 3.4. For each of the methods, only the top ten retrieved examples are shown. Qualitatively, better results are produced by the proposed method than the baselines as evidenced from the retrieved examples.



Query action: “applauding”

Query action: “pushing-a-cart”

Query action: “using_a_computer”

Figure 3.4: Top ten retrieval results (from left to right) of the proposed approach and the baseline methods for three different queries on the Stanford 40 actions dataset [37]. For each query, first row and second row refer to the retrieval results of the two baseline methods of NN_{Gen} and $\text{SVM}_{\text{multi}}$, respectively, while the third row refers to the retrieval results of the proposed approach NN_{ROC}. Images bounded in green boxes indicate relevant examples, while those bounded in red boxes are irrelevant.

Chapter 4

Optimizing IoU for Object Category Segmentation

In this chapter, we describe the proposed approach to address the object category segmentation problem by directly optimizing the intersection-over-union (IoU) performance measure in a deep learning framework. We give an approximation to the IoU loss and then directly incorporate it into the learning objective of a deep fully convolutional network.

4.1 Methodology

We consider here the problem of object category segmentation. Given an object category, the goal is to label the pixels of an image as being part of an object (foreground) of the category or not (background). To this end, we convert a classification CNN into a fully-convolutional CNN as proposed in [22], and then train the deep network end-to-end and pixel-to-pixel with an objective to directly optimize the IoU performance measure. The architecture of the deep network as well as details of the IoU loss function are discussed in the following subsections.

4.1.1 Network Architecture and Work Flow

Following the recent work for semantic segmentation by Long et al. [22], we start with a classification CNN called AlexNet [20], and replace the last two fully connected layers (fc_6 and fc_7) with 1x1 convolution layers (C_6 and C_7, respectively) to convert the CNN into a fully-convolutional network (FCN). We then add a scoring layer (C_8) which is also a 1x1 convolution layer. The sub-sampled output out of the scoring layer is then passed to a deconvolution layer (DC) that performs bilinear interpolation at a stride of 32 and produces an output equal to the size of the original input to the network. Up to this point, everything remains the same as the original 32 stride version of the FCN called "FCN-32s" proposed in [22].

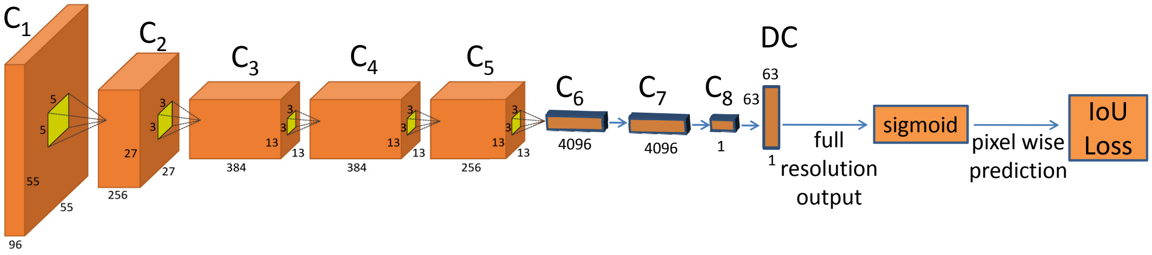


Figure 4.1: Architecture of the proposed FCN. The first eight convolution layers (C_1 – C_8) and the deconvolution layer (DC) remain the same as the original FCN-32s proposed in [22]. For each layer, the number right at the bottom represents the depth, while the other two numbers represent the height and width of the layer output. The yellow boxes inside C_1 – C_5 represent the filters, while the numbers around them represent filter dimensions. The IoU loss layer at the end computes IoU loss on the full-resolution output representing object class probabilities of the pixels.

Once an output equal to the size of the input is produced, we pass it through a sigmoid layer to convert the scores into class probabilities representing the likelihood of the pixels being part of the object. From this point forward, the proposed approach differs from [22], which computes softmax loss on each pixel score and trains the whole network based on this loss. We argue that this is not the right approach for a task like object category segmentation, where the ratio of object to background pixels is very small. The softmax loss is closely tied to the overall classification accuracy. If the number of examples in each class are balanced, minimizing the softmax loss will give high overall classification accuracy. For object category segmentation, the two classes are often very imbalanced, and therefore, the overall accuracy is not often a good performance measurement. For example, if 90% of the pixels belong to the background, a naive algorithm can achieve 90% overall classification accuracy simply by labeling every pixel as the background. In object category segmentation, the IoU score is often used as the standard performance measure, which takes into account of the class imbalance issue. Following this observation, instead of computing softmax loss, we pass the pixel probabilities out of the sigmoid layer to a loss layer that directly computes the IoU loss over all pixels in the training set and then train the whole FCN based on this loss. Figure 4.1 illustrates the architecture of the proposed network.

4.1.2 Approximation to IoU and IoU Loss

The IoU score is a standard performance measure for the object category segmentation problem. Given a database of images, the IoU measure gives the similarity between the predicted region and the ground-truth region for an object present in all

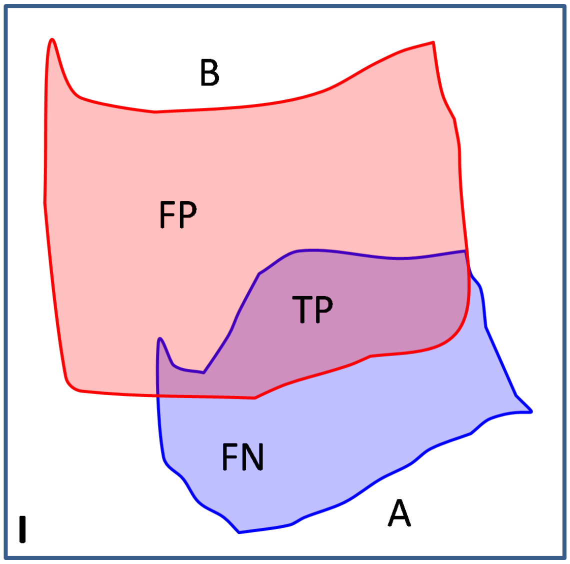


Figure 4.2: Visualization of the IoU metric.

or some of the images in the database. This is illustrated with the help of an example as shown in Fig. 4.2. Suppose, an image database includes only a single image I as shown in Fig. 4.2. Let A be an object present in I whose spatial extent is denoted by the region enclosed by the blue curve, whereas, the region enclosed by the red curve as denoted by B be the predicted region for the object. Then, the IoU metric can be defined by the following equation.

$$IoU = \frac{A \cap B}{A \cup B} = \frac{TP}{FP + TP + FN} \tag{4.1}$$

where, TP, FP, and FN denote the true positive, false positive and false negative counts, respectively.

From Eq. 4.1, we see that IoU score is a count based measure, whereas, the outputs of the proposed FCN are probability values representing likelihood of the pixels being part of the object. Therefore, we cannot measure the IoU score directly from the output of the network. We propose to approximate the IoU score using the probability values. More formally, let $V = \{1, 2, \ldots, N\}$ be the set of all pixels of all the images in the training set that is input to the network, X be the output of the network (out of the sigmoid layer) representing pixel probabilities over the set V, and $Y \in \{0, 1\}^V$ be the ground-truth assignment for the set V, where 0 represents background pixel and 1 represents object pixel. Then, the intersection $I(X)$, union $U(X)$ and the IoU count can be approximated as follows:

$$I(X) = \sum_{v \in V} X_v * Y_v \tag{4.2}$$

$$U(X) = \sum_{v \in V} (X_v + Y_v - X_v * Y_v) \tag{4.3}$$

$$IoU = \frac{I(X)}{U(X)} \tag{4.4}$$

Therefore, the IoU loss can be defined as:

$$L_{IoU} = 1 - IoU = 1 - \frac{I(X)}{U(X)} \tag{4.5}$$

We then incorporate this IoU loss L_{IoU} into the objective function of the proposed FCN, which takes the following form:

$$\arg\min_{w} L_{IoU} = 1 - IoU \tag{4.6}$$

where, w is the set of parameters of the deep network.

In order to obtain the optimal set of parameters w, Eq. 4.6 is solved using stochastic gradient descent. The gradient of the objective function with respect to the output of the network can then be written as follows:

$$\begin{aligned}\frac{\partial L_{IoU}}{\partial X_v} &= -\frac{\partial \frac{I(X)}{U(X)}}{\partial X_v} \\ &= \frac{-U(X) * \frac{\partial I(X)}{\partial X_v} + I(X) * \frac{\partial U(X)}{\partial X_v}}{U(X)^2} \\ &= \frac{-U(X) * Y_v + I(X) * (1 - Y_v)}{U(X)^2}\end{aligned} \tag{4.7}$$

which can be further simplified as follows:

$$\frac{\partial L_{IoU}}{\partial X_v} = \begin{cases} -\frac{1}{U(X)} & \text{if } Y_v = 1 \\ \frac{I(X)}{U(X)^2} & \text{otherwise} \end{cases} \tag{4.8}$$

Once the gradients of the objective function with respect to the network output is computed, we can simply backpropagate the gradients using the chain rule of derivative in order to compute the derivatives of the objective function with respect to the network parameters w.

4.2 Datasets

To evaluate the proposed approach, we conducted experiments on three different datasets – PASCAL VOC 2010 [10] and PASCAL VOC 2011 [11] segmentation datasets, as well as the Cambridge-driving Labeled Video Database (CamVid) [6].

4.2.1 PASCAL VOC 2010 and 2011 Segmentation Datasets

The PASCAL VOC segmentation datasets include high-resolution images of 20 different object categories along with their pixel-level annotations. The 2010 version of the dataset contains 964 training and 964 validation images, while the 2011 version includes 1,112 training and 1,111 validation images. We conducted training of the proposed approach and the baseline methods on 80% of the training data and used the remaining 20% training data for validation. We evaluated the different approaches on the validation set rather than the test set.

4.2.2 CamVid

CamVid is a road scene understanding dataset containing road scene videos taken from the perspective of a driving automobile. It includes over 10 minutes of high quality footage, and also provides 701 high resolution images extracted from the video sequences and the pixel-level semantic segmentations of the images. There are 11 different semantic object categories including “Road”, “Car”, “Building”, “Column-Pole”, “Sign-Symbols”, “Pedestrian”, “Fence” etc. Among the 701 images, 367 images are used for training, 233 for testing and the remaining 101 images are used for validation.

4.3 Experimental Setup

The focus of our work is object category segmentation. Therefore, for all the datasets, we conducted training on individual object categories and learned segmentation models for each object category separately. In other words, when we train on a particular object category, say dog, we assume pixels of all other categories as part of the background. During inference, we pass all test images through the learned models one for each object category, and then segment the specific objects individually from the test images. Details of the different baselines as well as the training setup are discussed in the following subsections.

4.3.1 Baselines

The hypothesis behind this work is that, for object category segmentation, learning a deep network using an application specific loss (in this case, IoU loss) is expected to produce better results than one using general classification loss. Therefore, as a primary baseline, we compare our proposed approach to a method proposed in [22] that uses general classification loss, more specifically, softmax loss for semantic segmentation. Moreover, we also compare our approach to a method proposed in [27] that also directly optimizes the IoU performance measure, but is based on structured Markov Random Filed (MRF) formulation. The model [27] is a linear model, not a deep model. The improvement over [27] will demonstrate the superiority of deep models over shallow models. For the rest of the thesis, we refer to the proposed deep model directly optimizing for IoU as FCN_{IoU}, the deep model optimizing for overall accuracy using softmax loss as FCN_{acc}, and the other MRF-based shallow model directly optimizing for IoU as MRF_{IoU}.

4.3.2 Training

Because batch gradient descent is slower as it performs gradient update on the whole training set, and because stochastic gradient descent fluctuates a lot as it performs gradient update on each example, we conducted training of FCN_{IoU} and FCN_{acc} using stochastic gradient descent in mini batches which draws the best of the two approaches. While preparing the mini batches, we made it sure that each mini batch contains at least one positive example (i.e., an image containing the object for which model is being trained), as the IoU measure is not defined when there is no positive example in the set. Both FCN_{IoU} and FCN_{acc} were initialized with pre-trained weights from AlexNet [20]. For the PASCAL datasets, we resized the training images to 375x500 for the sake of batch training, while testing was done on the original images without resizing. On the other hand, for the CamVid dataset, all the images were resized to 360x480. We used a fixed learning rate of 10^{-4}, momentum of 0.99 and weight decay of 0.0005. We continued training until convergence when there was no further improvement in the training loss and we chose the model with the best IoU measure on the validation set. We implemented the deep nets using a popular deep learning tool called MatConvNet [36]. Figure 4.3 shows a sample training curve for the proposed approach.

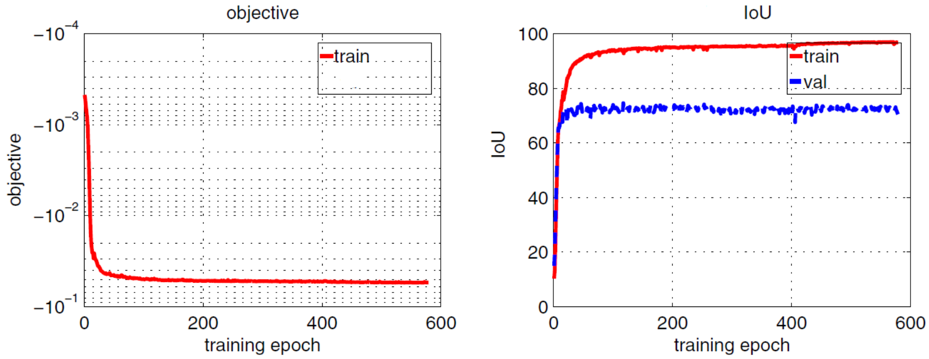


Figure 4.3: Sample training curve for the proposed approach.

4.4 Results

We report the results of the proposed approach and the baselines on different datasets in the following subsections.

4.4.1 PASCAL VOC 2010

Table 4.1: Intersection-over-union (%) performance comparison on the validation set of PASCAL VOC 2010 dataset [10] for 6 different object categories. FCN_{IoU} outperforms MRF_{IoU} on all categories, while performing better than FCN_{acc} on all but one category. Particularly noteworthy are the significant performance improvements for the categories with a relatively higher background to object pixel ratio as shown in Table 4.2.

Method	Aeroplane	Bus	Car	Horse	Person	TV/Monitor
MRF_{IoU}	<20	<30	<30	<10	<25	<15
FCN_{acc}	71.07	72.85	71.67	60.46	**75.42**	64.03
FCN_{IoU}	**75.27**	**74.47**	**72.83**	**61.18**	72.65	**67.37**

For the PASCAL VOC 2010 dataset [10], Table 4.1 shows the results of the proposed approach and the baselines on 6 different object categories, namely, "Aeroplane", "Bus", "Car", "Horse", "Person" and "TV/Monitor". The results on MRF_{IoU} are taken from [27]. Our proposed approach outperforms MRF_{IoU} by huge margin on all 6 categories. This performance boost is simply due to the powerful deep features learned automatically by the proposed approach FCN_{IoU}. In contrast, MRF_{IoU} is a shallow model and lacks the ability to learn features automatically. Please note that we could not report the exact IoU values of MRF_{IoU}, since [27] uses a bar chart to report the results and the exact numbers are not available in [27]. So we only report

Table 4.2: Background to object pixel ratio in PASCAL VOC 2010 [10] and 2011 [11] datasets.

Dataset	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	TV/Monitor
VOC 2010	152	319	107	142	150	64	66	40	97	152	82	75	117	91	25	182	111	99	85	104
VOC 2011	153	341	100	158	152	60	68	41	94	160	82	71	127	86	23	176	115	88	76	113

Table 4.3: Intersection-over-union (%) performance comparison on the validation set of PASCAL VOC 2011 [11]. FCN_{IoU} performs better than FCN_{acc} in most cases. Like PASCAL VOC 2010, performance improvements are more pronounced for categories with a relatively larger background to object pixel ratio as shown in Table 4.2.

Method	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Motorbike	Person	Potted Plant	Sheep	Sofa	Train	TV/Monitor	Mean
FCN_{acc}	72.18	60.57	66.47	64.68	**65.03**	73.96	71.82	71.44	55.55	**64.22**	62.74	**67.03**	60.74	70.23	**76.78**	61.62	67.59	58.05	72.80	65.05	63.18
FCN_{IoU}	**75.07**	**62.00**	**67.45**	**67.64**	65.00	**75.37**	**72.87**	**71.94**	**56.01**	64.13	**63.91**	65.71	**60.92**	**70.90**	73.61	**63.78**	**68.83**	**58.56**	**72.66**	**66.81**	**63.82**

the upper bounds for MRF_{IoU}.

While comparing the proposed approach FCN_{IoU} to the primary baseline FCN_{acc} on the PASCAL VOC 2010 dataset [10], we see that FCN_{IoU} outperforms FCN_{acc} in almost all categories, except the "Person" category. It is particularly noteworthy that the performance improvements are more pronounced for object categories (e.g., "Aeroplane", "TV/Monitor" etc.) where the ratio of the background to object pixels is very large as shown in Table 4.2.

4.4.2 PASCAL VOC 2011

For the PASCAL VOC 2011 dataset [11], we report results of FCN_{IoU} and the primary baseline FCN_{acc}, as the other baseline does not report any results on this dataset. Table 4.3 shows the results on all 20 object categories of the PASCAL VOC segmentation dataset. The proposed approach performs better than the baseline in most cases. Specifically, the performance improvement is more pronounced for object categories with a larger ratio of background to object pixels.

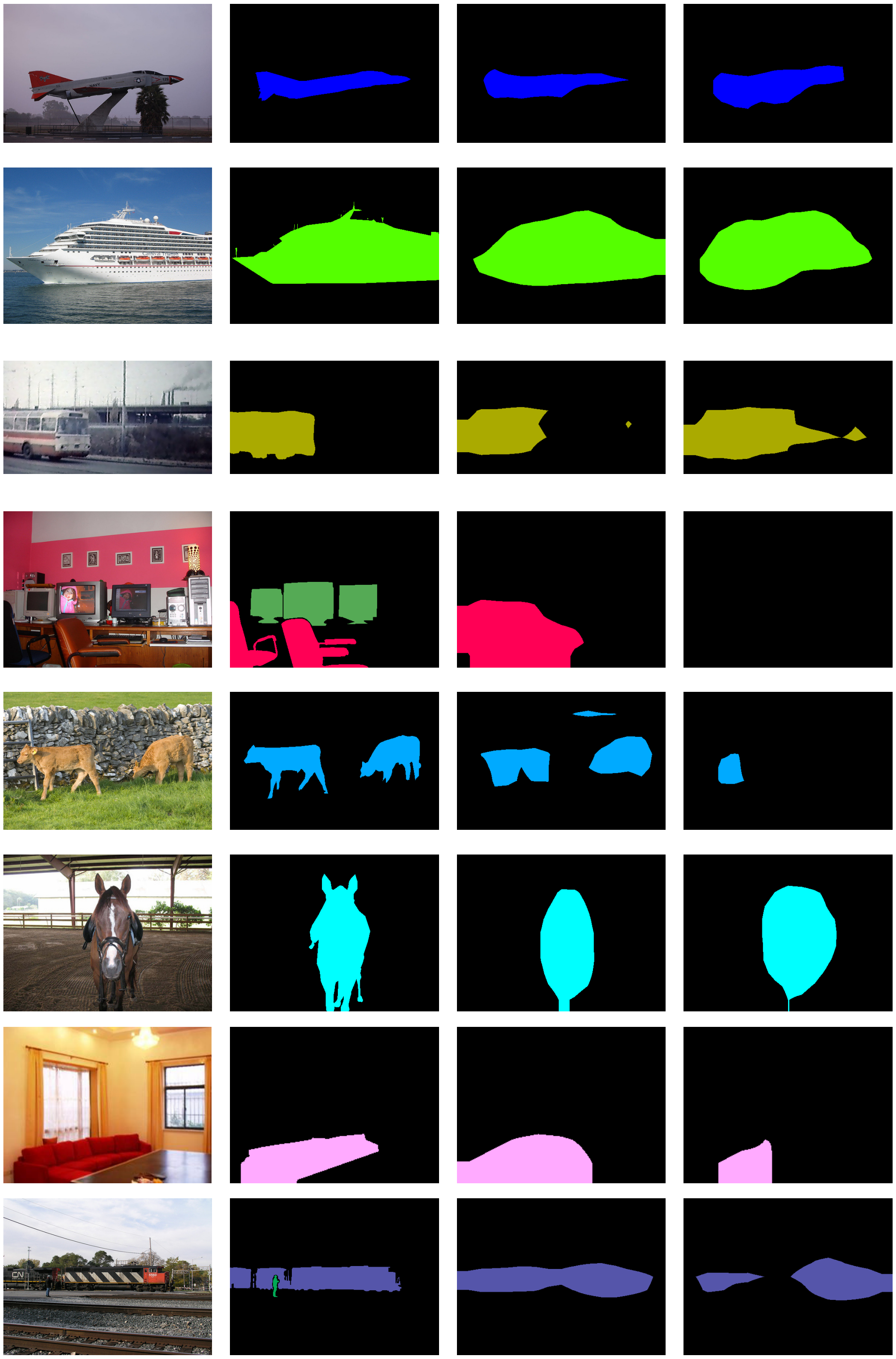


Figure 4.4: Sample segmentations on the PASCAL VOC 2011 validation set [11]. Columns (left to right): original images, ground-truth segmentations, segmentations produced by $\mathrm{FCN}_{\mathrm{IoU}}$, and segmentations produced by $\mathrm{FCN}_{\mathrm{acc}}$.

We also show some qualitative results of the proposed approach FCN_{IoU} and the primary baseline FCN_{acc} in Fig. 4.4. Since the softmax loss used in FCN_{acc} is tied to the overall classification accuracy, the FCN_{acc} model tends to misclassify object pixels as background (i.e., false negative), since there are more background pixels. In contrast, FCN_{IoU} directly optimizes the IoU score, so the model tends to recover some of the false negative errors made by FCN_{acc}.

4.4.3 CamVid

For the CamVid dataset, [6], we report results on 5 categories: “Road”, “Building”, “Column-Pole”, “Sign-Symbol”, and “Fence”. We choose the “Road” and “Building” categories for their high ratio of background to object pixels, while the other categories are chosen for the opposite reason. Figure 4.5 shows the data distri-

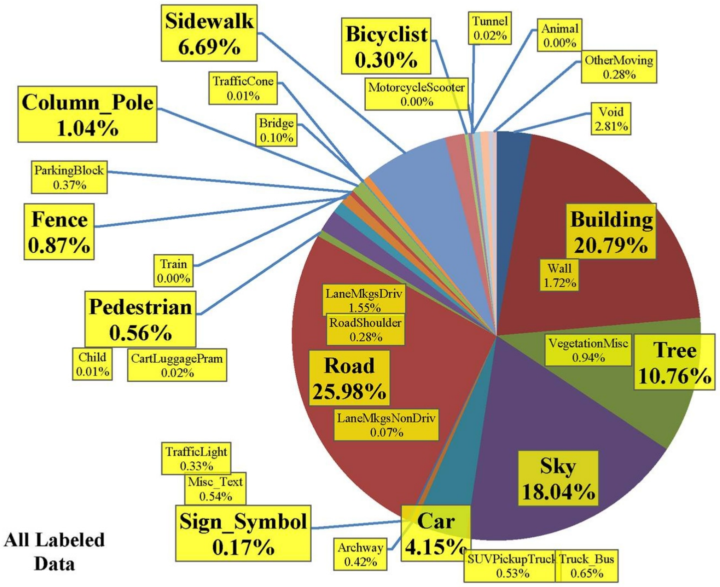


Figure 4.5: Data distribution of the CamVid dataset [6]. Figure taken from [1].

Table 4.4: Intersection-over-union (%) performance comparison on the CamVid dataset [6] for 5 different object categories. FCN_{IoU} performs better than FCN_{acc} on all categories in both validation and test sets. Performance improvements are more pronounced for smaller object categories.

Method	Road		Building		Column-Pole		Sign-Symbol		Fence	
	val	test	val	test	val	test	val	test	val	test
FCN_{acc}	95.53	90.38	87.03	76.21	50.46	50.91	64.94	56.27	75.97	61.75
FCN_{IoU}	**95.58**	**90.69**	**88.30**	**76.72**	**53.48**	**52.79**	**67.78**	**57.78**	**80.68**	**62.23**

bution of the CamVid dataset. The IoU scores on the 5 object categories are reported in Table 4.4. The results show that FCN_{IoU} outperforms FCN_{acc} in all 5 categories.

As with the PASCAL dataset, we also show some qualitative results on the CamVid dataset [6] in Fig. 4.6. The results show that FCN_{IoU} performs better than FCN_{acc}, specially for the smaller object categories (e.g., Column-Pole) where there exists huge imbalance in the number of object and background pixels.

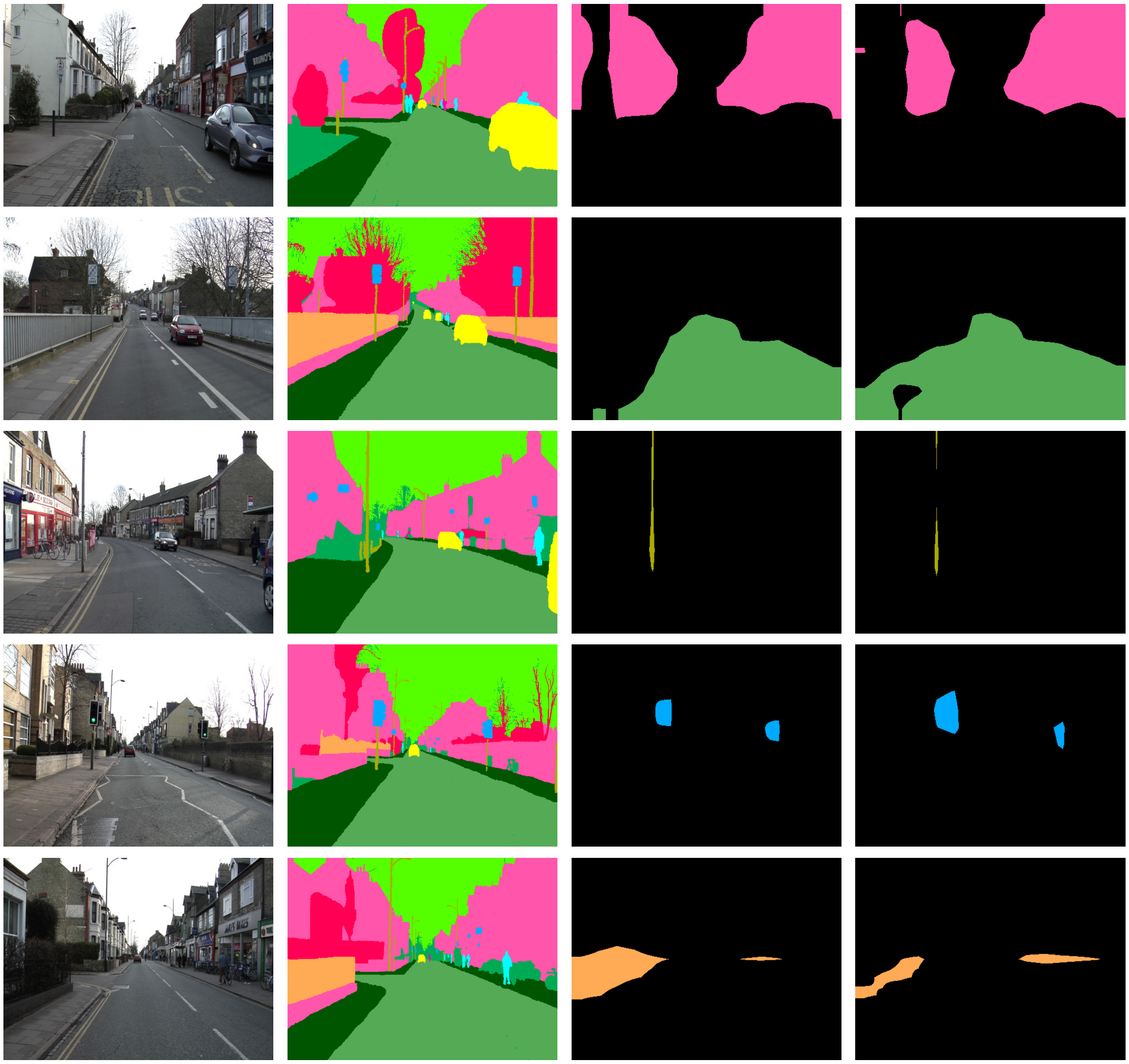


Figure 4.6: Sample segmentations on the CamVid dataset [6]. Rows (top to bottom): segmentations for "Building", "Road", "Column-Pole", "Sign-Symbol", and "Fence". Columns (left to right): original images, ground-truth segmentations, segmentations produced by FCN$_{\text{IoU}}$, and segmentations produced by FCN$_{\text{acc}}$.

Chapter 5

Conclusion

In this thesis, we studied the problem of application specific performance measure optimization in a deep learning setting. We particularly addressed the action retrieval and the object category segmentation problems by directly optimizing the performance measures ROC area and IoU, respectively using deep learning. To this end, we provided approximation to ROC area loss and IoU loss and then incorporated these loss functions into the learning objectives of the respective deep networks. We also validated the superiority of our proposed approach over different baselines through extensive experiments on several benchmark datasets.

Possible future directions of research regarding the thesis are listed below:

- For the action retrieval problem, we directly optimize for ROC area. But, ROC area is mainly used in a retrieval setting that considers only binary relevance. We, therefore, aim to extend this work to be able to handle multi-level relevance by directly optimizing for performance measures like Normalized Discounted Cumulative Gains [16].

- The proposed method for directly optimizing IoU can only deal with object to background segmentation. It cannot handle segmentations of all object categories simultaneously, as doing so would require optimizing a sum of several fractions (category specific IoU measures), which itself is a very hard optimization problem. Therefore, it would be really interesting to explore on optimizing such functions in future work.

Bibliography

[1] Camvid data distribution. `http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/`.

[2] Ahmed, F., Tarlow, D., and Batra, D. Optimizing expected intersection-over-union with candidate-constrained crfs (2015). In *International Conference on Computer Vision (ICCV 2015)* (2015).

[3] Bai, Y., Yang, K., Yu, W., Ma, W., and Zhao, T. Learning high-level image representation for image retrieval via multi-task DNN using clickthrough data. *CoRR abs/1312.4740* (2013).

[4] Bai, Y., Yu, W., Xiao, T., Xu, C., Yang, K., Ma, W.-Y., and Zhao, T. Bag-of-words based deep neural network for image retrieval. In *Proceedings of the ACM International Conference on Multimedia* (2014), MM '14.

[5] Blaschko, M. B., and Lampert, C. H. Learning to localize objects with structured output regression. In *Proceedings of the 10th European Conference on Computer Vision: Part I* (2008), ECCV '08, pp. 2–15.

[6] Brostow, G. J., Fauqueur, J., and Cipolla, R. Semantic object classes in video: A high-definition ground truth database. *Pattern Recogn. Lett.* (2009), 88–97.

[7] Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., and Fei-Fei, L. Imagenet large scale visual recognition competition 2012 (ILSVRC2012), 2012.

[8] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR abs/1310.1531* (2013).

[9] Du, T., Lubomir, B., Rob, F., Lorenzo, T., and Manohar, P. Learning spatiotemporal features with 3d convolutional networks. *arXiv preprint arXiv:1412.0767* (2015).

[10] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[11] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[12] Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. Simultaneous detection and segmentation. In *European Conference on Computer Vision (ECCV)* (2014).

[13] Hariharan, B., Arbeláez, P., Girshick, R., and Malik, J. Hypercolumns for object segmentation and fine-grained localization. In *Computer Vision and Pattern Recognition (CVPR)* (2015).

[14] Hazan, T., Keshet, J., and McAllester, D. A. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems 23*. 2010, pp. 1594–1602.

[15] Herschtal, A., and Raskutti, B. Optimising area under the roc curve using gradient descent. In *Proceedings of the Twenty-first International Conference on Machine Learning* (2004), ICML.

[16] Järvelin, K., and Kekäläinen, J. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2000), SIGIR '00.

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

[18] Joachims, T. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[19] Joachims, T. A support vector method for multivariate performance measures. In *Proceedings of the 22Nd International Conference on Machine Learning* (2005), ICML '05, pp. 377–384.

[20] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25* (2012), pp. 1097–1105.

[21] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing* (2001).

[22] Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (2015).

[23] Mcfee, B., and Lanckriet, G. Metric learning to rank. In *In Proceedings of the 27th annual International Conference on Machine Learning (ICML)* (2010).

[24] Ng, J. Y.-H., Yang, F., and Davis, L. S. Exploiting local features from deep networks for image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (2015).

[25] Nowozin, S. Optimal decisions from probabilistic models: The intersection-over-union case. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* (2014).

[26] Premachandran, V., Tarlow, D., and Batra, D. Empirical minimum bayes risk prediction: How to extract an extra few% performance from vision models with just three more parameters. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (June 2014).

[27] Ranjbar, M., Lan, T., Wang, Y., Robinovitch, S., and Mori, G. Optimizing non-decomposable loss functions in structured prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2012).

[28] Razavian, A. S., Sullivan, J., Maki, A., and Carlsson, S. A baseline for visual instance retrieval with deep convolutional networks. *CoRR abs/1412.6574* (2014).

[29] Sharma, A., Tuzel, O., and Liu, M.-Y. Recursive context propagation network for semantic scene labeling. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on* (2015).

[30] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR abs/1409.1556* (2014).

[31] Song, Y., Schwing, A. G., Zemel, R. S., and Urtasun, R. Direct loss minimization for training deep neural nets. In *Proceedings of the 33rd International Conference on Machine Learning* (2016), ICML '16.

[32] Soomro, K., Zamir, A. R., and Shah, M. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR abs/1212.0402* (2012).

[33] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. *CoRR abs/1409.4842* (2014).

[34] Tarlow, D., and Zemel, R. S. Structured output learning with high order loss functions. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012* (2012), pp. 1212–1220.

[35] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res. 6* (dec 2005).

[36] Vedaldi, A., and Lenc, K. Matconvnet – convolutional neural networks for matlab. *CoRR abs/1412.4564* (2014).

[37] Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L. J., and Li, F.-F. Human action recognition by learning bases of action attributes and parts. In *ICCV'11* (2011), pp. 1331–1338.

[38] Yue, Y., Finley, T., Radlinski, F., and Joachims, T. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (2007), SIGIR '07.