Spring 2014

Incorporating Boltzmann Machine Priors for Semantic Labeling in Images and Videos

Andrew Kae
University of Massachusetts - Amherst

INCORPORATING BOLTZMANN MACHINE PRIORS FOR SEMANTIC LABELING IN IMAGES AND VIDEOS

A Dissertation Presented

by

ANDREW KAE

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2014

Computer Science

INCORPORATING BOLTZMANN MACHINE PRIORS FOR SEMANTIC LABELING IN IMAGES AND VIDEOS

A Dissertation Presented

by

ANDREW KAE

Approved as to style and content by:

Erik Learned-Miller, Chair

Allen Hanson, Member

Benjamin Marlin, Member

John Staudenmayer, Member

Lori Clarke, Department Chair
Computer Science

To my family.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people that have helped and encouraged me throughout my graduate study.

First of all, I would like to thank my advisor, Erik Learned-Miller, for all his support and guidance through the years. In addition, I would like to thank my fellow students in the Vision Lab for their support and friendship: Moe Mattar, Gary Huang, Laura Sevilla, Jacqueline Feild, Manjunath Narayana, Vidit Jain, Jerod Weinman, Ben Mears, David Smith and Adam Williams.

Next, I would like to thank faculty members that I have had the privilege of knowing, including Allen Hanson, Ben Marlin, David Smith, R. Manmatha, and James Allen. I would also like to thank my past collaborators: Kihyuk Sohn, Honglak Lee, and Carl Doersch. In addition, I thank my colleagues Kin Kan and Vijay Narayanan, for a wonderful summer at Yahoo! Labs. I also had a great experience the following summer at Osaka Prefecture University working with Koichi Kise and Masakazu Iwamura.

Lastly, I thank my family for supporting me throughout this long journey.

ABSTRACT

INCORPORATING BOLTZMANN MACHINE PRIORS FOR SEMANTIC LABELING IN IMAGES AND VIDEOS

MAY 2014

ANDREW KAE

B.A., CORNELL UNIVERSITY

M.Eng., CORNELL UNIVERSITY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Erik Learned-Miller

Semantic labeling is the task of assigning category labels to regions in an image. For example, a scene may consist of regions corresponding to categories such as sky, water, and ground, or parts of a face such as eyes, nose, and mouth. Semantic labeling is an important mid-level vision task for grouping and organizing image regions into coherent parts. Labeling these regions allows us to better understand the scene itself as well as properties of the objects in the scene, such as their parts, location, and interaction within the scene. Typical approaches for this task include the conditional random field (CRF), which is well-suited to modeling local interactions among adjacent image regions. However the CRF is limited in dealing with complex, global (long-range) interactions between regions in an image, and between frames in a video. This thesis presents approaches to modeling long-range interactions within images and videos, for use in semantic labeling.

In order to model these long-range interactions, we incorporate priors based on the restricted Boltzmann machine (RBM). The RBM is a generative model which has demonstrated the ability to learn the shape of an object and the CRBM is a temporal extension which can learn the motion of an object. Although the CRF is a good baseline labeler, we show how the RBM and CRBM can be added to the architecture to model both the *global* object shape within an image and the *temporal dependencies* of the object from previous frames in a video. We demonstrate the labeling performance of our models for the parts of complex face images from the Labeled Faces in the Wild database (for images) and the YouTube Faces Database (for videos). Our hybrid models produce results that are both quantitatively and qualitatively better than the baseline CRF alone for both images and videos.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

Segmentation and semantic labeling are core techniques for the critical mid-level vision tasks of grouping and organizing image regions into coherent parts. Segmentation refers to the grouping of image pixels into parts without assigning labels to those parts, and semantic labeling assigns specific category names to those parts. By grouping and organizing regions in an image, we gain a better understanding not only of *what* objects are in an image but also their *context* and how they interact with one another.

This thesis presents work to segment and label face scenes as an intermediate step to modeling face structure. By better understanding the face structure, we can describe the face in terms of high-level features or *attributes* [51, 70] as well as potentially improve performance in related tasks such as face recognition. There is much practical value in improved performance for these tasks in applications such as image retrieval, surveillance, and photo-tagging.

Specifically, we address the problem of labeling face regions in images and videos with hair, skin, and background labels. Our work is primarily involved with labeling, and not segmentation. Thus, each face image is first pre-segmented into superpixel regions [71, 66, 12] before our model assigns labels to the regions. Table 1.1 shows an example of a face image, its superpixel segmentation and its corresponding ground truth labeling. Similarly, for videos, each video frame is pre-segmented into superpixel regions before our model assigns labels to the regions. Table 1.2 shows frames

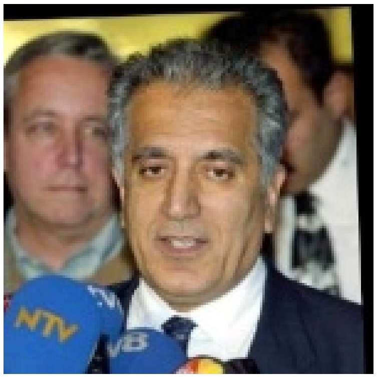
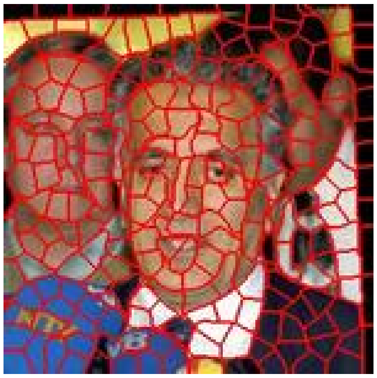
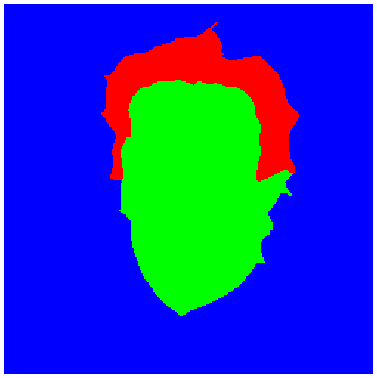
Aligned Image	Superpixel	Ground Truth
		

Table 1.1. The left image shows a "funneled" or aligned image using the method of Huang et al. [43], taken from the Labeled Faces in the Wild (LFW) database [46]. The center image shows the superpixel segmentation of the image which is used as a basis for labeling. The right image shows the ground truth labeling. **Red** represents hair, **green** represents skin, and **blue** represents background.

from a face video with the corresponding superpixel segmentations and ground truth labelings.

The particular task of Hair/Skin/Background labeling serves as an ideal domain in which to evaluate our models. It is a more constrained problem compared to labeling general scenes since we assume there is a centered face that has already been cropped out and roughly aligned before running our models. In contrast, for general scenes the problem is less constrained since there can be multiple objects in varying locations present within a scene. By focusing on the more constrained problem first, we can work on building and evaluating our models without additional complication. We can then focus on extending our models to the less constrained problem afterward as future work. Even though the task of Hair/Skin/Background labeling is a more constrained problem, it is still a difficult problem, due to the variety of poses, hair and skin shapes present in faces. In addition, there are complicated part relationships present, such as between hair shape and pose (for example, a person facing to the left will have less visible hair on their left side). We also show in Chapter 4 that our

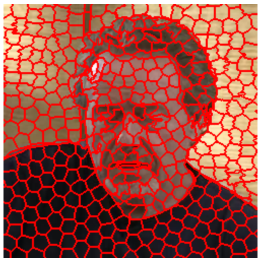
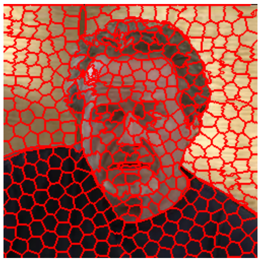
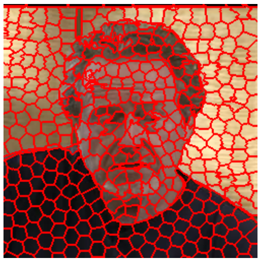
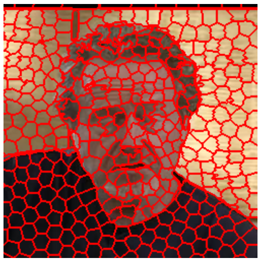
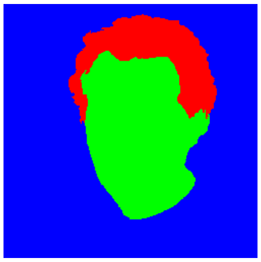
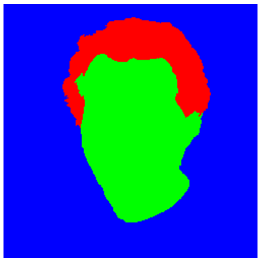
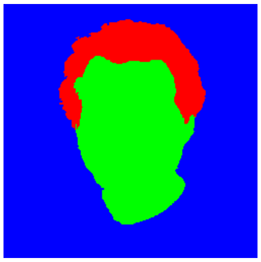
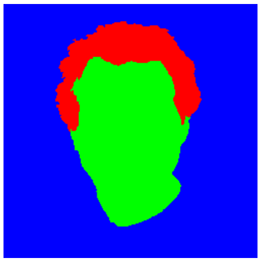
	t	**t+2**	**t+4**	**t+6**
YFDB				
Superpixel				
Ground Truth				

Table 1.2. The first row shows every other frame from a video in the Youtube Faces Database (YFDB) [97]. The second rows shows the temporal superpixel segmentation and the last row shows ground truth. **Red** represents hair, **green** represents skin, and **blue** represents background.

model can learn simple attributes such as the pose of the face and hair length, which may be useful for tasks such as retrieval.

For semantic labeling applications, the conditional random field (CRF) [52] is widely used since it is effective at modeling region boundaries as shown in [80, 45, 27]. For our task, the CRF can model a correct transition between the hair and background labels when there is a clear difference between those regions. However, when a person's hair color is similar to that of the background, the CRF may have difficulty deciding where to place the boundary between the regions.

In such cases, a *global* shape constraint can be used to filter out unrealistic label configurations. It has been shown that the restricted Boltzmann machine (RBM) [82] and its extension to deeper architectures such as the deep Boltzmann machine (DBM) [74], can be used to build effective generative models of object shape. Specifically, the shape Boltzmann machine (ShapeBM) [20] showed impressive performance in generating novel and realistic object shapes while capturing both local and global elements of object shape.

Motivated by these examples, we propose the *GLOC* (GLObal and LOCal) model as a strong model for image labeling problems, that combines the best properties of the CRF (that enforces *local consistency* between adjacent regions) and the RBM (that models the *global shape prior* of the object). The model balances three goals in seeking label assignments:

- Region labels should be consistent with the underlying image features.
- Region labels should respect image boundaries.
- The complete image labeling should be consistent with shape priors learned from labeled training data.

In our GLOC model, the first two objectives are achieved primarily by the CRF component, and the third objective is addressed by the RBM component. For each new image, our model uses mean-field approximate inference to find a good balance between the CRF and RBM potentials in setting the image labels and hidden node values.

For videos, a traditional CRF can be extended to include temporal potentials from previous frames [92, 27], but it may difficult to model higher order temporal and shape dependencies. For example, if a person is moving their head toward the right, we would like the model to capture the shape and temporal dependencies involved with this motion. In order to incorporate these dependencies into a CRF framework, we present the *STRF* (Shape-Time Random Field), as a strong model for video labeling problems. In this model, the prior takes the form of a conditional restricted Boltzmann machine (CRBM) [86] which is a temporal extension to the RBM. The STRF model also uses mean-field approximate inference to efficiently find a balance between the CRF and CRBM potentials.

The GLOC model is evaluated on a subset of images from the Labeled Faces in the Wild (LFW) database [46] and the STRF model is evaluated on a subset of videos from the YouTube Faces Database (YFDB)[97]. In both cases, these models offer significant improvements in labeling accuracy (both qualitatively and quantitatively) over baseline methods, such as the CRF. These gains in numerical accuracy have a significant visual impact on the resulting labeling, often fixing errors that are small but obvious to an observer.

The main contributions of this thesis are as follows:

- GLOC, a strong model for the face labeling task in images. GLOC combines CRF and RBM components to model both local and shape consistency.
- STRF, a strong model for the face labeling task in videos. STRF combines CRF and CRBM components to achieve local, shape, and temporal consistency.
- Efficient inference and training algorithms for both GLOC and STRF.
- GLOC and STRF outperform competitive baselines both qualitatively and quantitatively.
- GLOC can learn face attributes automatically without attribute label supervision.
- Both the code [1] and labeled data [2] used for GLOC are publicly available. The code and labeled data used for STRF will be made available upon publication.

The rest of this thesis is organized as follows. Section 1.1 reviews related work in semantic labeling for images and videos. Chapter 2 reviews prior work in modeling object shape. Chapter 3 provides background material on the CRF and RBM components used in the GLOC and STRF models. Chapter 4 presents the GLOC model and chapter 5 presents the STRF model. Finally, Chapter 6 reviews the work presented and concludes the thesis.

1.1 Related Work

This section reviews related work in segmentation and region labeling in both images and videos. Recall that segmentation is the task of grouping image pixels into parts without applying labels to those parts, and region labeling assigns specific category labels to those parts (such as "Sky" or "Ground"). We first review work in segmentation and labeling in general scenes and then focus on face scenes in particular.

1.1.1 General Scene Segmentation

Segmentation is a core problem in computer vision that has been applied to a variety of tasks such as tracking, recognition, and region labeling. In the early 20th century *Gestalt* psychologists such as Wertheimer [95] hypothesized that we understand a scene as a whole rather than as a sum of its parts. They identified several factors that humans use to group objects together, such as similarity, proximity, and continuity. These ideas have had a large influence on the approaches used for segmentation in computer vision. Based on these principles, a good segmentation should partition the image into smooth, spatially contiguous regions that are uniform with respect to appearance features such as color or texture.

1.1.1.1 Images

Perhaps the simplest segmentation technique is to threshold the image based on pixel values (such as Otsu's method [68]), resulting in a binary image. Thresholding (and its variations such as adaptive thresholding and multi-level thresholding) may work for very controlled settings such as fingerprint scans but for complex, natural scenes which may contain many changes in illumination, thresholding is often not ideal.

Many segmentation algorithms are based on a top-down splitting of regions or a bottom-up merging of regions, including superpixels [71, 66, 12], which is used in our work. One of the first algorithms used for this top-down splitting approach is based on the quadtree [40], in which an initial root node contains the entire image and nodes are then split or merged according to the homogeneity of the pixels within a node. Region-growing algorithms [34] are an example of the bottom-up class of approaches in which each pixel is represented as a node in a graph and edges are added between nodes if the pixels are similar in appearance.

Clustering-based approaches have also been used for segmentation. K-means is a simple clustering algorithm that can be used to segment images based on features such as color or texture, but this approach has the disadvantage of knowing K, the number of clusters, beforehand. An alternative approach is to use the mean-shift [28] algorithm, a non-parametric approach to finding modes of a distribution from data samples. This technique has been applied to segmentation [14] by assigning each pixel to its closest mode, which has the effect of clustering together pixels with similar appearance. However, this approach is known to be computationally expensive.

Superpixels [71, 66, 5] have emerged as a popular mid-level representation between low-level pixels and larger scene segments. Superpixels group together pixels that share similar visual characteristics and act as atomic subregions of the image. Superpixels can then replace pixels as the base representation in an image which can reduce computational complexity significantly since there are typically many fewer superpixels than pixels in an image. Table 1.1 shows an example of an image and its corresponding superpixel representation. Superpixels can be generated using normalized cuts [79], a graph-based approach which partitions the graph (where nodes correspond to pixels) depending on the similarity between pixels in an image.

There has also been interesting work in interactive image segmentation. In this setting, a user helps guide the segmentation in ways such as (1) selecting a sample of pixels belonging to the foreground and a sample of pixels belonging to the background [9], or (2) by drawing a bounding box around the object of interest [72]. In some cases, there may be multiple iterations of this user-guided segmentation. While this type of interactive segmentation may be useful in applications such as photo-editing, it is not practical to expect this type of extra supervision when dealing with very large image databases, such as found currently online.

A related problem to image segmentation is matting, which is the task of foreground extraction in images. It is commonly used in image and film editing for applications such as moving the foreground object into another scene. The matting problem was introduced by Porter & Duff [69] and the goal is to estimate an α value which ranges from $\{0, 1\}$ for every pixel. A value of 0 indicates that a pixel is definitely background and a value of 1 indicates that the pixel is definitely foreground. Thus, matting can be seen as a continuous version of the segmentation problem discussed so far. Regarding our task of face labeling, while it may be useful for labels to have a continuous value, it also makes it more difficult to learn and evaluate a model since the correct label value α may be ambiguous. In addition the labeling task may be further complicated when considering multiple category labels, such as in our task.

1.1.1.2 Videos

Segmentation in videos has important applications to tasks such as activity recognition, surveillance and tracking. The approaches can be roughly divided into whether frames are processed in an online fashion or whether all frames in a video are processed together. For example, the work of Dementhon [19] required all frames to be segmented together using a mean-shift clustering approach. Later, Grundmann [32] proposed a hierarchical segmentation system but still required all frames to be segmented together. In contrast, works such as Vasquez-Reina et al. [87] present an online video segmentation system similar in spirit to the P^n models of Kohli et al. [49] for image labeling. They use multiple segmentations per frame and aggregate these guesses to generate a final set of segmentations.

In addition, there has also been work in extending the mid-level representations of superpixels from images to videos, such as supervoxels [99, 100] and temporal superpixels (TSP) [12]. Supervoxels were introduced as the extension of the superpixel for videos and 3D volumetric data (such as found in medical imaging). TSPs [12] were introduced recently as potentially more appropriate for video data than supervoxels, since they tend to maintain better label consistency and are more uniform in size than many supervoxel approaches. In this thesis, we use TSPs for segmentation because they are processed in an online manner (i.e. in real time) in contrast to most supervoxel algorithms (with the exception of [100]) and offer better segmentation performance.

1.1.2 General Scene Labeling

Labeling is the task of assigning categories (such as “Sky” or “Ground”) to image regions. In general, the goal is to not only identify the objects (or semantic regions) in a scene but also the *context* in which these objects/regions interact and thereby better understand what is going on in the scene. Labeling is a critical sub-task of this larger goal since it requires both identifying the objects and also their segmentations.

An example of scene labeling is shown in Figure 1.1, taken from the LabelMe [73] database. LabelMe [73] is a publicly available database providing roughly labeled segmentations for various outdoor and indoor scenes. By labeling a scene such as in Figure 1.1, we can learn useful part relationships about objects such as the fact

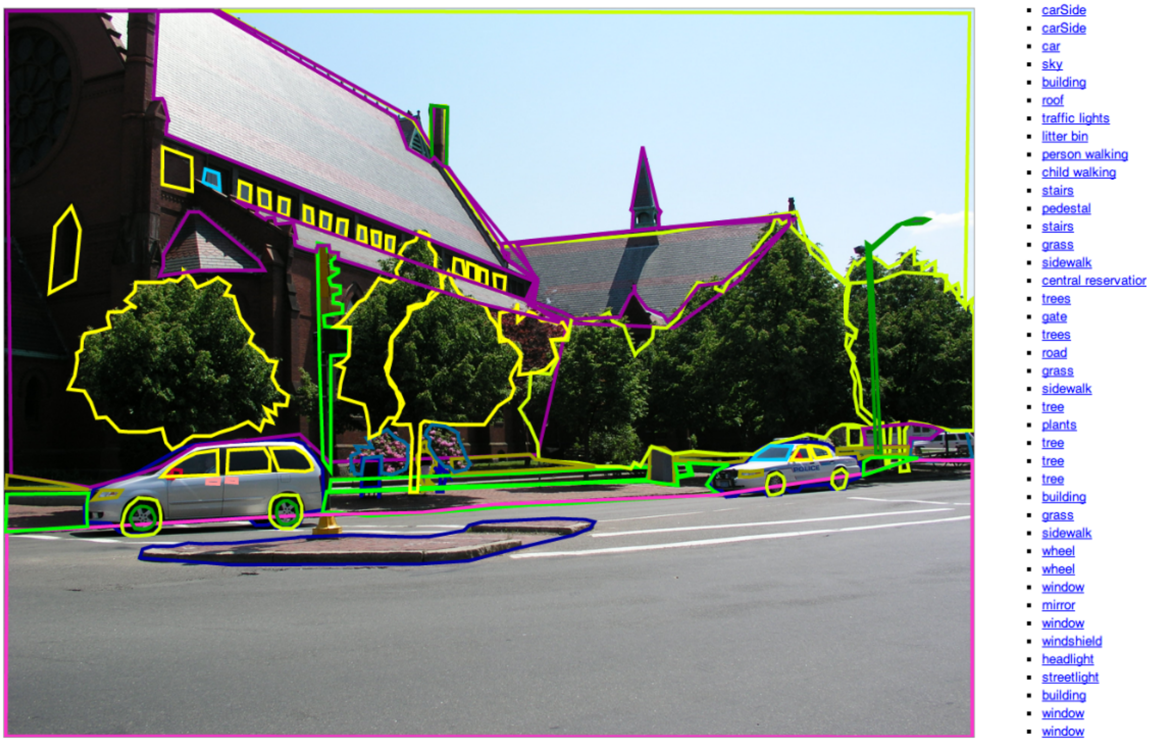


Figure 1.1. Scene Labeling. An example image from the LabelMe [73] database. The scene is segmented into rough regions corresponding to category labels (given on the right) such as "Sky", "Trees", or "Sidewalk."

that cars contain both wheels and windows as subparts, and that buildings can also contain windows. We can also infer spatial relationships between objects such as the fact that cars are typically found on a road, or that "Sky" is typically above the road. Knowing these relationships can help object detection for an outdoor scene as shown in Figure 1.1. For example in a scene with cars, road and sky, it would be strange to also detect a computer or a table. Finally, knowing these relationships can also help to classify the scene itself. Knowing that cars are on the road can help to determine that the image is an outdoor street scene.

1.1.2.1 Images

There has been much work in labeling regions into categories such as "Sky" or "Ground" [80, 31, 49, 36, 33] for images. Early work includes the VISIONS [33] system which not only labeled regions but performed scene analysis including depth layers. Yakimovsky et al. [102] also developed a semantic labeling system for scenes using a region-growing algorithm and knowledge of the scene.

One common approach to semantic labeling is to model the image using a Markov Random Field (MRF) where nodes correspond to image regions (or pixels) and edges are connected between adjacent regions in the image. Some approaches [80, 31] model the pixels directly whereas others first segment the image into regions such as superpixels and then assign labels to each region such as in [45]. Typically, a disadvantage of modeling pixels directly in a grid-structured MRF is that for even moderately sized images, the number of pixels is very large which leads to a very large graph, complicating inference. The main disadvantage of modeling at the superpixel level is that it is possible that there could be errors in the segmentation, such as multiple classes within the same superpixel. In this thesis, we use superpixels primarily for the reduced computational complexity. We show later in Chapter 4 that the error associated with having multiple labels within a superpixel is not large.

Some approaches incorporate global scene information for use in labeling. Gould et al. [31] added geometric constraints into their model to capture spatial information such as: “Sky” is above “Ground”. In addition, there have been several works that incorporated higher order potentials in a CRF framework for image labeling. He et al. [36] proposed multiscale CRFs to model both local and global label features using RBMs. Specifically, they used multiple RBMs at different scales to model the regional or global label fields (layers) separately, and combined those conditional distributions multiplicatively. Our model is similar in that we also use an RBM as a global shape model, but our work differs in that we include edge potentials for local smoothness between adjacent label nodes. We also make our model computationally efficient by defining it on superpixels. The P^n models [49] also incorporate higher order potentials into a CRF by using multiple segmentations per image to improve labeling performance. However, they do not use any global shape information.

1.1.2.2 Videos

Semantic labeling in videos is related to problems such as object tracking [15] and background subtraction [83], in which a foreground object (or multiple objects) is extracted from the background in a video. There has also been work in the semi-supervised task of label propagation in videos. For example, Badrinarayanan et al. [7] proposed a semi-supervised system to propagate labels for use in labeling road scenes, given an initial labeling.

Some works have extended the use of MRFs and CRFs from images to videos, for semantic labeling. For example, Wang et al. [92] incorporates temporal potentials from previous frames in a video. Wojek et al. [4] incorporates a higher order temporal potential using the output of an object detector in order to jointly detect and label objects in a scene. For our task, we assume that the object of interest (a face) has already been cropped out as a pre-processing step, but incorporating an object detector may be more appropriate for general scene labeling. Floros et al. [27] presented a system to semantically label road scenes by first pre-segmenting the video frames into superpixels and using 3D point correspondences as a higher level potential. If a point matches another point in 3D space, then they should be linked together. The main difference between this approach and ours is that they lacks a global shape model. In addition, our work also incorporates temporal potentials similar to [92].

1.1.3 Face Scene Labeling

The following section covers both segmentation and labeling for face scenes.

1.1.3.1 Images

Several authors have built systems for labeling hair, skin, and other face parts [90, 89, 77, 55, 101, 45]. Because of the variety of hair styles, configurations, and amount of hair, the shape of a hair segmentation can be extremely variable. In our work, we treat facial hair as part of "hair" in general, hoping to develop hidden units corresponding to beards, sideburns, mustaches, and other hair parts, which further increases the complexity of the hair segments. Furthermore, we include skin of the neck as part of the "skin" segmentation when it is visible, which is different from other labeling regimes. For example, Wang et al. [89] limit the skin region to the face and include regions covered by beards, hats, and glasses as being skin, which simplifies their labeling problem.

Yacoob et al. [101] build a hair color model and then adopt a region growing algorithm to modify the hair region. This method has difficulty when the hair color changes significantly from one region to another (especially for dark hair), and their work was targeted at images with controlled backgrounds. Lee et al. [55] used a mixture model to learn six distinct hair styles, and other mixture models to learn color distributions for hair, skin, and background.

Huang et al. [45] used a standard CRF trained on images from LFW to build a hair, skin, and background labeler. We have implemented their model as a baseline and report the performance in Chapter 4. Scheffler et al. [77] learn color models for four classes: hair, skin, background and clothing. They also learn a spatial prior for each class label and then combine this information with a MRF that enforces local label consistency.

Wang et al. [89] used a compositional exemplar-based model, focusing mostly on the problem of hair segmentation. Following their earlier work, Wang et al. [90] proposed a model that regularizes the output of a segmentation using parts. In addition, their model builds a statistical model of *where* each part is used in the image and the co-occurrence probabilities between parts. Using these co-occurrences, they build a tree-structured model over parts to constrain the final segmentations. To our knowledge, this was the best-performing algorithm for hair, skin, and background labeling at the time we developed our own models. In Chapter 4, we report the results showing improvements over their best results.

There has also been work to perform finer grained face segmentation such as the LabelFaces model [93], which labels subparts such as eyes and nose. In addition, Luo et al. [62] developed a system to automatically parse faces in a hierarchical manner using a deep belief net (DBN) [39].

1.1.3.2 Videos

In face scene videos, there has been work to extract the face regions [11, 58], which can be useful for face recognition applications. Some works segment the face, neck and shoulder regions together [57], which is is similar to our task since we consider face and neck to be part of the skin category. There has also been working in applying some of the work in contour models towards segmentation of face parts. For example, Lievin et al. [61] built a multi-stage system that first roughly segments the face, then specific parts such as lips and eyes. A contour-based model is then used to achieve a more fine-grained segmentation. However, due to the sequential nature of the model, if the initial face detection is incorrect, then the part-based segmentation will also be incorrect.

Overall, while there has been work which treats face regions as foreground objects to be extracted from video, there has generally been less work towards the semantic labeling of face parts, which is one of the tasks covered in this thesis. This is an important problem because it has applications to tasks such as surveillance in which we may be interested in finding the person with red hair among a group of people in a video.

1.2 Discussion

This chapter has introduced the problem of semantic labeling for both general scenes and face scenes, in images and videos. Overall, while there has been much work done in scene labeling, there has not been much work in incorporating a strong *global* shape prior for labeling. We will demonstrate the utility of this kind of shape prior for improving the performance of a traditional CRF model for face scene labeling for both images (in Chapter 4) and videos (in Chapter 5).

Next, chapter 2 will review models of object shape before describing our models in more detail. Chapter 3 will cover the components of our models and then Chapters 4 and 5 will present our models for the semantic labeling of images and videos, respectively, of face scenes.

CHAPTER 2

OBJECT SHAPE MODELING

One of the distinguishing features of our models compared to others used for semantic labeling is the incorporation of a strong, global model of object shape, which is used to complement the local features within a discriminative model. This chapter reviews and compares several approaches to modeling object shape, including the restricted Boltzmann machine (RBM), which is the shape model used in this thesis.

2.1 MRF Based Models

One common approach to model the shape of an object is to use a Markov random field (MRF) for use in computer vision tasks such as image denoising and segmentation [30, 59]. A simple form of an MRF is the Ising model which consists of binary variables (and the more general Potts model which consists of multinomial variables). The Ising model is typically defined on a grid or lattice structured graph in which local interactions are modeled in the form of unary and pairwise potentials. One limitation of the Ising model in modeling shapes is that it only considers local interactions and typically does not generate realistic looking shapes [67].

Boykov et al.[9] present an interactive segmentation system in which a user selects a sample of background pixels and foreground pixels. Their model then learns to separate the foreground object from the background by performing graph cuts on an MRF based on pixels. While their model does learn to segment the foreground object, it requires user interaction (and potentially multiple iterations) in order to obtain a good segmentation. In contrast, once our models are learned, labeling over new images is fully automatic.

The restricted Boltzmann machine (RBM) [82] is a type of MRF which is structured as a bipartite, undirected graph, with a layer of visible units and hidden units. The RBM is the shape prior used in our work and will be covered in detail in Chapter 3.3. There have been several works which use the RBM (or models based on the RBM) as a shape model. For example, He et al. [36] used an RBM to model both local and global label object shape within a scene. Specifically, they used multiple RBMs at different scales to capture local and global label fields separately, and combined those conditional distributions multiplicatively. Local label fields learn about interactions of specific objects within the scene while global label fields learn more general scene properties, such as the fact that the sky should be at the top of the image. Our model differs in that we include edge potentials for local smoothness between adjacent label nodes. We also make our model computationally efficient by defining it on superpixels, and then map superpixels into the visible units of the RBM using the novel concept of a virtual visible layer (more details in Chapter 4).

Recent work by Eslami et al. [20] introduced the Shape Boltzmann machine (SBM), which is a two-layer deep Boltzmann machine (DBM)[74] with local connectivity in the first layer for local consistency and generalization (by weight sharing), and full connectivity in the second layer for modeling global shapes. The SBM model showed impressive generative ability when it sampled realistic (binary) object shapes of animals such as horses and rhinos. In our work, we use the RBM which contains only a single layer of hidden units, instead of multiple layers like the DBM or SBM. In their paper, the RBM was found to have good overall generative performance, but lacks some of the fine details (such as the detailed tail and leg shapes of horses) captured by the SBM. However, it is simpler to train and perform inference on the

RBM, and so we decided to base our shape prior on the RBM rather than the more complicated SBM or DBM.

Subsequently, Eslami and Williams [21] proposed a generative model by combining the SBM with an appearance model for parts-based object segmentation. The main difference to our work is that we incorporate the RBM into a discriminative CRF framework and use the virtual pooling to map between superpixels and the fixed grid of the RBM, whereas their representation is based on pixels. Recently, Yujia et al. [60] proposed a similar idea to ours where they incorporate an RBM as a global shape prior into a CRF framework. Our work was published simultaneously and we were unaware of their work at the time. In addition, they also based based their image features at the pixel level, whereas we use superpixels, which can reduce computational complexity and simplify model inference.

2.2 Contour Based Models

A different approach to modeling object shape is to learn a contour model, such as an Active Shape Model (ASM) [17], which is a generic outline of an object that can iteratively deform to fit new examples. The ASM is defined over a set of landmark points placed around the contour (or outline) of an object, such as a hand. PCA is then used to find a basis for these points and then new examples are fit by finding suitable parameters for the learned basis. The Active Appearance Model (AAM) [16] model is an extension of the ASM which accounts for the appearance of the object in addition to the contour, resulting in a more robust model.

One disadvantage of this class of models is that they depend on having a training set with a large number of carefully placed landmark points on the contour of the object. These landmark points must correspond to the same positions, consistently, across all images. In some cases, it is clear where to place landmark points but in other cases, it can be ambiguous. For example, if we are modeling a hand, the tip of each finger may be an obvious landmark point, but it is unclear which points along the contours of the fingers should also be landmark points, and how many. Thus, this manual labeling is a time-consuming and somewhat ambiguous process, since it is unclear where to place some landmark points. In addition, contour-based models are limited in their ability to handle variations in shape.

Winn et al. [96] present a related model called LOCUS, an unsupervised model which can learn an object shape and perform object segmentation. They adapt to new images by finding parameters to deform a learned shape model, similar to the ASM [17]. The main advantage of this approach is that it is unsupervised and can learn to segment shapes if the objects are aligned. However, one disadvantage is that it may be difficult to model small variations well. For example, the variety of different hair shapes may be difficult to capture under this model.

2.3 Part Based Models

Another way to represent an object is through a collection of components or parts. Pictorial structures (PS) [26, 23] are a popular example of this class of approach, in which an object is modeled through a deformable configuration of rigid parts. The appearance of each part is modeled separately and the overall geometric arrangement of parts is based on a physical spring-like model. The PS allows for a wide range of motions through the deformable nature of the parts. For example, the human body may be modeled by a collection of simplified rectangles corresponding to parts such as the head, arms, legs, torso and then brought together by connecting parts such as arms and legs to the body. The spring-like connections between each pair of parts model realistic deformations such as the leg or arm being raised or bent. These PS models have been successfully used for tasks such as objection recognition [23].

In addition, Objcut [50] is a system that augments the local information from a CRF with global shape information using PS. For our task, PS may be unsuitable since it may not cover all the variations present in the face label shapes. For example, it is possible that some parts (such as Hair) may be absent in some images and it is unclear how the PS can account for missing parts.

A related model is the deformable part model, introduced by Felzenswalb et al. [24, 22], which is conceptually similar to pictorial structures (PS). However, instead of the rigid parts used in PS, deformable part models rely on part templates based on HOG [18] features. In addition, the system uses a coarse template (also based on HOG features) to capture global object shape. Deformable part models are currently popular models used for for object detection [22] and pose estimation [103]. One disadvantage of this style of approach is that it requires a user-defined global template whereas we would prefer our model to learn the shape and its parts automatically.

Another related model is the constellation model [10, 94, 25] in which an object is represented by a constellation of parts. Here, a part refers to the output of a feature detector, with an associated location, scale, and appearance vector. The overall shape of an object is represented by a Gaussian distribution over the positions of the detected features. While the constellation model may be appropriate to use for object detection, it may be less appropriate for labeling because the parts are the outputs of a feature detector and it is unclear how to use this model for fine-grained labeling. In addition, the parts do not correspond to semantic parts, but rather points that are found by a feature detector, and so they may lack a semantic meaning. As presented later in Chapter 4, the shape prior we use can have semantic meanings or attributes.

2.4 Template Based Models

Another class of models is based on finding matches to an existing collection of shapes or templates in order to learn the structure of the object. For example, Borenstein et al [8], represent an object class (such as horses), by image patches or fragments, which are then used later to match to a new image. Chen et al. [13] learn a dictionary of *shape epitomes* which are mid-level edge representations of shape, that are incorporated into a CRF for labeling. Gavrila et al. [29] has worked on an exemplar-based approach to shape modeling in which they build a tree of shape templates through a hierarchical clustering of training shapes. New objects are detected by searching through the tree, comparing the template at each node until a “match” is found at a leaf node. One disadvantage of this class of approaches is that it lack a global shape model, which can be useful for labeling.

2.5 Discussion

This chapter covered several approaches to modeling object shape and discussed their advantages and disadvantages. The shape model used in this thesis is the restricted Boltzmann machine (RBM). As mentioned by Eslami et al. [20], the RBM was able to learn and generate novel, realistic label shapes of objects such as horses. While it may not capture the fine details that a deeper model can (such as their shape Boltzmann machine), it is simpler to train and perform inference on the RBM. Therefore, we decided to use the RBM as our shape model because it offers a good tradeoff between generative ability and ease of use.

To justify the RBM as the object shape model for our task of Hair/Skin/Background labeling, an RBM was trained on the labeled face images in our training set (consisting of 1500 examples). Figure 2.1 shows generated samples from the RBM in the top row and their closest matching training examples in the bottom row.[1] The generated samples are clearly different from the closest training example and so the RBM is not just “memorizing” training examples. The RBM is learning meaningful structures such as hair and beard shapes, as well as their co-occurrences (i.e. we do not observe

[1] The L_2 distance between a generated sample and the training examples is used to find the closest match.

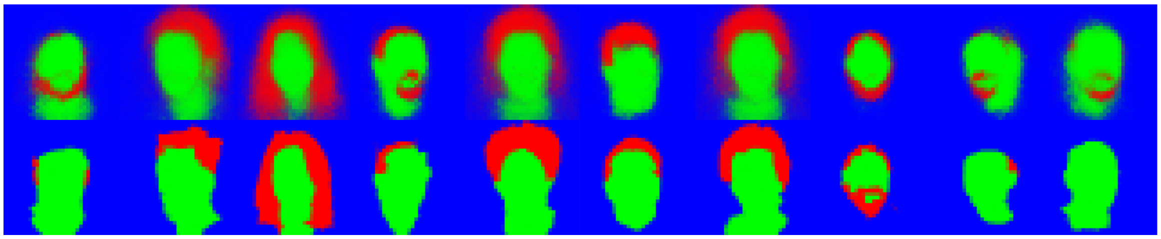


Figure 2.1. Generated samples from the RBM (first row) and the closest matching examples in the training set (second row). The RBM can generate novel, realistic examples by combining hair, beard and mustache shapes along with diverse face shapes.

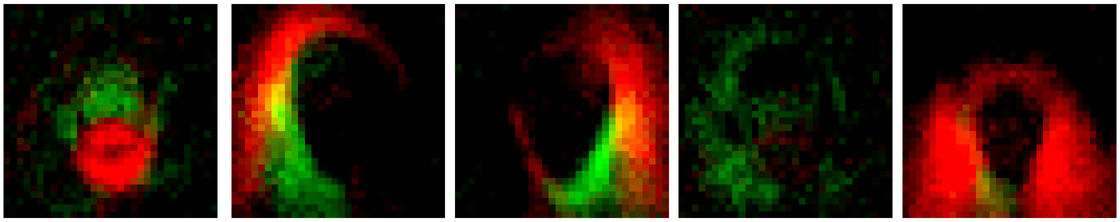


Figure 2.2. RBM hidden unit visualizations. This figure shows the pairwise weights for 5 particular hidden units. Weights for the hair label are shown in **red**. Weights for the skin label are shown in **green**. Background weights are set to zero by default. Each filter shown is 32 × 32 pixels.

women with long hair and beards). Importantly, the RBM is able to learn a variety of face and hair shapes that *look realistic*.

In addition to the samples, we can visualize the RBM hidden units weights to determine whether the RBM is learning a meaningful structure. Figure 2.2 shows these hidden unit weights (also called filters) for five particular hidden units. Weights for the hair label are shown in **red** and weights for the skin label are shown in **green**. Background weights are set to zero by default. Each filter shown in the figure is 32×32 pixels. It is clear that the filters correspond to learned structures (or parts) present in the face shapes. For example, the first filter seems to correspond to a "Beard" part. The second and third filters appear to correspond to a person facing right and then left, respectively. The fourth filter may correspond to an absence of hair and the last filter may correspond to a person with a large amount of hair. Figures 2.1

and 2.2 indicate that the RBM has learned a strong model of face shape since it can sample realistic face shapes and the learned filters correspond to meaningful face structures or parts. More detail about the structure and formulation of the RBM will be covered in Chapter 3. Chapters 4 and 5 will show how to use the RBM for labeling face regions in images and videos.

CHAPTER 3
ALGORITHMS

This chapter reviews the conditional random field (CRF) and restricted Boltzmann machine (RBM) models which are the components used in our proposed models. In particular, this chapter covers formulations of these models along with learning and inference using these models for the labeling task. Chapter 4 presents the proposed GLOC model (GLObal and LOCal) which incorporates the CRF and RBM for semantic labeling in images. Chapter 5 presents the the proposed STRF model (Shape-Time Random Field), which incorporates the CRF and conditional restricted Boltzmann machine (CRBM), used for semantic labeling in videos.

Notation. Let us define the set of variables used throughout this thesis.

- An image I is pre-segmented into $S^{(I)}$ superpixels, where $S^{(I)}$ can vary over different images. The superpixels correspond to the nodes in the graph for image I.
- Let $\mathcal{G}^{(I)} = \{\mathcal{V}^{(I)}, \mathcal{E}^{(I)}\}$ represent the nodes and edges for the undirected graph of image I.
- Let $\mathcal{V}^{(I)} = \{1, \cdots, S^{(I)}\}$ denote the set of superpixel nodes for image I.
- Let $\mathcal{E}^{(I)} = \{(i,j) : i,j \in \mathcal{V}^{(I)}$ and i,j are adjacent superpixels in image $I\}$[1].
- Let $\mathcal{X}^{(I)} = \{\mathcal{X}_{\mathcal{V}}^{(I)}, \mathcal{X}_{\mathcal{E}}^{(I)}\}$ be the set of features in image I, where

[1] Adjacent superpixels share a common boundary.

  - $\mathcal{X}_{\mathcal{V}}^{(I)}$ is the set of node features $\{\mathbf{x}_s^{\text{node}} \in \mathbb{R}^{D_n}, s \in \mathcal{V}^{(I)}\}$ for image I.
  - $\mathcal{X}_{\mathcal{E}}^{(I)}$ is the set of edge features $\{\mathbf{x}_{ij}^{\text{edge}} \in \mathbb{R}^{D_e}, (i,j) \in \mathcal{E}^{(I)}\}$ for image I.

- Let $\mathcal{Y}^{(I)} = \{\mathbf{y}_s \in \{0,1\}^L, s \in \mathcal{V}^{(I)} : \sum_{l=1}^{L} y_{sl} = 1\}$ be the set of labels for the nodes in image I.

D_n and D_e denote the dimensions of the node and edge features, respectively, and L denotes the number of labels. In the rest of this section, the superscripts "I", "node", and "edge" are omitted for clarity, but the meaning should be clear from the context.

3.1 Conditional Random Field

The conditional random field [52, 84] is a powerful model for structured output prediction (such as sequence prediction [75] and text parsing [52, 78]) and has been widely used in computer vision [35, 6, 8, 36]. The CRF is a discriminative model which is structured as an undirected graphical model (or Markov network) in which the nodes are divided into a latent (or unobserved) set and an observed set. Usually, the goal is to infer the latent nodes given the observed nodes which are always conditioned on. The main advantage of a CRF over the simpler logistic regression (LR) model is that the CRF accounts for neighboring interactions among nodes. For example, CRFs can model edge interactions such as neighboring words in a sentence or neighboring pixels in an image.

The conditional distribution and the energy function for the CRF in the labeling task are defined as follows:

$$P_{\text{crf}}(\mathcal{Y}|\mathcal{X}) \propto \exp(-E_{\text{crf}}(\mathcal{Y}, \mathcal{X})), \tag{3.1}$$

$$E_{\text{crf}}(\mathcal{Y}, \mathcal{X}) = E_{\text{node}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}\right) + E_{\text{edge}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}\right), \tag{3.2}$$

$$E_{\text{node}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}\right) = -\sum_{s \in \mathcal{V}} \sum_{l=1}^{L} \sum_{d=1}^{D_n} y_{sl} \Gamma_{ld} x_{sd}, \tag{3.3}$$

$$E_{\text{edge}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}\right) = -\sum_{(i,j) \in \mathcal{E}} \sum_{l,l'=1}^{L} \sum_{e=1}^{D_e} y_{il} y_{jl'} \Psi_{ll'e} x_{ije}, \tag{3.4}$$

where $\Gamma \in \mathbb{R}^{L \times D_n}$ represent the node weights and $\Psi \in \mathbb{R}^{L \times L \times D_e}$ is a 3D tensor for the edge weights.

For the labeling task covered in this thesis, the CRF can model node features at either the pixel or superpixel level and model edge features for additional smoothing between nodes. In this thesis, nodes are represented at the superpixel level and not at the pixel level for two reasons. First, the superpixel representation is computationally much more efficient. The images in our database are of size 250 × 250, which (if modeled at the pixel level) corresponds to a graph of 250^2 nodes, which may be too large to perform efficient approximate inference. However, each image can be segmented into a much smaller number of about 200-250 superpixels. The graph for the CRF can then be modeled at this superpixel level (where nodes correspond to superpixels), allowing for a more efficient approximate inference. Second, superpixels can help smooth features such as color. For example, if a superpixel consists mostly of black pixels but contains a few interspersed blue pixels, the blue pixels will be smoothed out from the feature vector, which may help simplify inference.

Also, note that E_{edge} is computed using edge features (similar to the TextonBoost model [80]), which is slightly different than the typical computation of E_{edge} which does not include edge features. By incorporating edge features, the model can perform a dynamic smoothing which depends on the underlying features, rather than just smoothing the labels. This encourages the model to assign the same label to neighboring superpixel nodes if the underlying superpixels are similar in appearance.

3.1.1 Inference

In this thesis, the nodes in the CRF are modeled at the superpixel level with edges connecting nodes if the corresponding superpixels are adjacent in the image. In general, this results in a loopy graph and so an approximate inference is necessary. There are three commonly used approaches for approximate inference in Markov networks: Markov-chain Monte Carlo (MCMC), variational approaches, and loopy belief propagation. We briefly review these approaches and then provide more detail about the specific approximate inference approach used in our experiments, the mean-field approximation.

MCMC approaches are based on drawing many samples from a Markov chain that is used to approximate the distribution of interest. In the limit, these samples will eventually be drawn from the true posterior distribution. This class of approximate inference methods includes the widely used Gibbs sampler and Metropolis-Hastings sampler. Variational approaches treat the approximate inference task as an optimization problem, which minimizes the difference between the true posterior and an approximation. Typically, variational approaches are known to be faster than MCMC-based approaches while being less accurate. Lastly, loopy belief propagation (LBP) is simply the belief propagation algorithm for tree-structured graphs applied to general graphs which may contain loops. LBP may not always converge and if it does converge, the solution is generally not exact, but is still considered to be a good approximation.

In this thesis, we use the variational style of approximate inference because of the speed advantage over MCMC-based approaches. In particular, we use a simple version of variational approximation known as mean-field [76]. In this approximation, nodes are considered to be independent of one another. Even though this is a simple approximation, we have observed empirically that the mean-field approximation performed well. We decided to use the mean-field approximation instead of LBP because

Algorithm 1 Mean-Field inference for the CRF

```
1: Initialize $\boldsymbol{\mu}^{(0)}$ as follows:
       $$\mu_{sl}^{(0)} = \frac{\exp\left(f_{sl}^{\text{node}}\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}}\right)}$$
   where
       $$f_{sl}^{\text{node}}\left(\mathcal{X}_{\mathcal{V}}, \Gamma\right) = \sum_{d} x_{sd}\Gamma_{dl}$$
2: for i=0:$MaxIter$ (or until convergence) do
3:     update $\boldsymbol{\mu}^{(i+1)}$ as follows: $\mu_{sl}^{(i+1)} =$
           $$\frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right)\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right)\right)}$$
       where
           $$f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}; \mathcal{X}_{\mathcal{E}}, \mathcal{E}, \Psi\right) = \sum_{j:(s,j)\in\mathcal{E}} \sum_{l',e} \mu_{jl'}\Psi_{ll'e}x_{sje}$$
4: end for
```

mean-field is guaranteed to converge (typically to a local optimum) whereas LBP may not converge at all.

3.1.1.1 Mean-Field inference

The variational approach involves approximating the true posterior $P_{\text{crf}}(\mathcal{Y}|\mathcal{X})$ with a simpler graphical model, parameterized as $Q(\mathcal{Y};\mu)$. The goal is to update the parameters μ to make the approximation $Q(\mathcal{Y};\mu)$ as close as possible to the true posterior $P_{\text{crf}}(\mathcal{Y}|\mathcal{X})$, by minimizing the KL divergence $\text{KL}\left(Q(\mathcal{Y};\mu)\|P(\mathcal{Y}|\mathcal{X})\right)$. In the case of mean-field inference, the nodes are considered independently and so the approximation simplifies to the fully factorized distribution $Q(\mathcal{Y};\mu) = \prod_{s\in\mathcal{V}} Q(\mathbf{y}_s)$, with $Q(\mathbf{y}_s = l) \triangleq \mu_{sl}$.

The mean-field inference for our model formulation is shown in Algorithm 1. The variational parameters $\boldsymbol{\mu}_{sl}^{(i)}$ are essentially the current estimates for the posterior distribution for the nodes at step i. The variables f_{sl}^{node} and f_{sl}^{edge} correspond to the node and edge energies, respectively, at the node s with label l. Note that in Step 1, the variational parameters $\boldsymbol{\mu}_{sl}^{(0)}$ are initialized to the logistic regression guess, which relies only on node energies. The $MaxIter$ variable is the maximum the number of iterations to run for inference. Empirically, we observed that 200 is a good value for this upper limit. The algorithm loops until either this upper limit is reached or the variational parameters $\boldsymbol{\mu}$ no longer change with updates.

3.1.2 Learning

The model parameters $\{\Gamma, \Psi\}$ in Equations (3.3) and (3.4) are trained to maximize the conditional log-likelihood of the training data $\{\mathcal{Y}^{(m)}, \mathcal{X}^{(m)}\}_{m=1}^{M}$, assuming the training data consists of M examples. The log-likelihood of the data is defined as

$$L = \max_{\Gamma,\Psi} \sum_{m=1}^{M} \log P_{\text{crf}}(\mathcal{Y}^{(m)}|\mathcal{X}^{(m)}).$$

The gradient for the node weights Γ is defined as

$$\frac{\partial L}{\partial \Gamma_{dl}} = \frac{1}{M} \sum_{m=1}^{M} \left(\sum_{s\in\mathcal{V}} y_{sl} x_{sd} - \sum_{s\in\mathcal{V}} P(y_{sl}|x) x_{sd} \right),$$

and the gradient for the edge weights Ψ is defined as

$$\frac{\partial L}{\partial \Psi_{ll'e}} = \frac{1}{M} \sum_{m=1}^{M} \left(\sum_{(i,j)\in\mathcal{E}} y_{il} y_{jl'} x_{ije} - \sum_{(i,j)\in\mathcal{E}} P(y_{il}|x) P(y_{jl}|x) x_{ije} \right).$$

In both gradients, the negative component is obtained using the mean-field inference procedure in Algorithm 1. With the log-likelihood and gradient formulations given above, the model parameters $\{\Gamma, \Psi\}$ are learned using the limited-memory Broyden Fletcher Goldfarb Shanno (LBFGS) optimization algorithm in conjunction with the minFunc [3] software package.

3.1.2.1 Regularization

In many cases, to avoid overfitting, regularization is used during learning in order to obtain a better model. This typically involves adding an extra term to the log-likelihood optimization function, which penalizes large model weights. In the experiments, we tried several regularization values and chose the value which performed best on the validation set.

3.2 Spatial CRF

We now present a variant of the CRF that was found to work better in practice for labeling than the standard CRF[52, 84]. After the object in the image has been aligned to a canonical position (using an approach such as congealing [44, 43]), the image is divided into an $N \times N$ grid. The model then learns a separate set of node weights for each cell n in this grid while the edge weights are kept globally stationary. The node energy is revised to

$$E_{\text{node'}}(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}) = -\sum_{s \in \mathcal{V}} \sum_{l=1}^{L} y_{sl} \sum_{n=1}^{N^2} p_{sn} \sum_{d=1}^{D_n} \Gamma_{ndl} x_{sd}, \tag{3.5}$$

where $\Gamma \in \mathbb{R}^{N^2 \times D \times L}$ is a 3D tensor specifying the connection weights between the superpixel node features and labels at each spatial location. In this energy function, the projection matrix $\{p_{sn}\}$ specifies the mapping from the $N \times N$ grid to superpixels. The projection matrix $\{p_{sn}\}$ is defined as

$$p_{sn} = \frac{Area(Region(s) \cap Region(n))}{Area(Region(s))},$$

where $Region(s)$ denotes the set of pixels corresponding to superpixel s and $Region(n)$ denotes the set of pixels corresponding to grid position n. Details about how the CRF and SCRF are used in our experiments will be described in the next chapter.

Algorithm 2 Mean-Field inference for the SCRF

```
Initialize $\boldsymbol{\mu}^{(0)}$ as follows:
    $\mu_{sl}^{(0)} = \frac{\exp\left(f_{sl}^{\text{node}}\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}}\right)}$
  where
    $f_{sl}^{\text{node}}\left(\mathcal{X}_{\mathcal{V}}, \{p_{sn}\}, \Gamma\right) = \sum_{n,d} p_{sn} x_{sd} \Gamma_{ndl}$
for i=0:$MaxIter$ (or until convergence) do
    update $\boldsymbol{\mu}^{(i+1)}$ as follows: $\mu_{sl}^{(i+1)} =$
        $\frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(t)}\right)\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(t)}\right)\right)}$
      where
        $f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}; \mathcal{X}_{\mathcal{E}}, \mathcal{E}, \Psi\right) = \sum_{j:(s,j)\in\mathcal{E}} \sum_{l',e} \mu_{jl'} \Psi_{ll'e} x_{sje}$
end for
```

3.2.1 Inference

Inference in the SCRF is also done using mean-field as shown in Algorithm 2. The only difference is in the update for f_{sl}^{node} since it now include the projection matrix $\{p_{sn}\}$. Otherwise inference proceeds the same way as in the CRF.

3.2.2 Learning

The only difference between the CRF and SCRF is the node energy shown in Equation 3.5, and so the edge weights Ψ gradients remain the same as in the CRF. For the node weights Γ, the gradient is slightly modified to

$$\frac{\partial L}{\partial \Gamma_{ndl}} = \frac{1}{M} \left(\sum_{m=1}^{M} \sum_{s\in\mathcal{V}} x_{sd} y_{sl} p_{sn} - \sum_{s\in\mathcal{V}} x_{sd} P(y_{sl}|x) p_{sn} \right), \tag{3.6}$$

which now includes the projection matrix $\{p_{sn}\}$. As before with the CRF, the negative component is approximated using mean-field inference as shown in Algorithm 2.

The parameters are learned using LBFGS optimization in conjunction with the minFunc [3] software package.

3.3 Restricted Boltzmann Machine

The restricted Boltzmann machine (RBM) [82] is a bipartite, undirected graphical model composed of visible and hidden layers as shown in Figure 3.1. It is called "restricted" due to the bipartite nature of the graph, in contrast to a more general Boltzmann machine [38] which allows intra-layer connections. In our context of image labeling, there are R^2 multinomial visible units $\mathbf{y}_r \in \{0,1\}^L$ and K binary hidden units $h_k \in \{0,1\}$. The joint distribution is defined as

$$\begin{aligned} P_{\text{rbm}}(\mathcal{Y},\mathbf{h}) &\propto \exp(-E_{\text{rbm}}(\mathcal{Y},\mathbf{h})), \qquad (3.7)\\ E_{\text{rbm}}(\mathcal{Y},\mathbf{h}) &= -\sum_{r=1}^{R^2}\sum_{l=1}^{L}\sum_{k=1}^{K} y_{rl}W_{rlk}h_k \\ &\quad -\sum_{k=1}^{K} b_k h_k - \sum_{r=1}^{R^2}\sum_{l=1}^{L} c_{rl}y_{rl}, \qquad (3.8)\end{aligned}$$

where $\mathbf{W} \in \mathbb{R}^{R^2\times L\times K}$ is a 3D tensor specifying the connection weights between visible and hidden units, b_k is the hidden bias, and c_{rl} is the visible bias.

3.3.1 Inference

Inference in the RBM can be done efficiently by taking advantage of the conditional independence structure of the graph. Each hidden unit is conditionally independent of the other hidden units given the visible units and similarly, each visible unit is conditionally independent of the other visible units given the hidden units. During inference, we can perform a block Gibbs sampling in which all the hidden units are sampled together given the visible units and then all the visible units are sampled together given the hidden units.

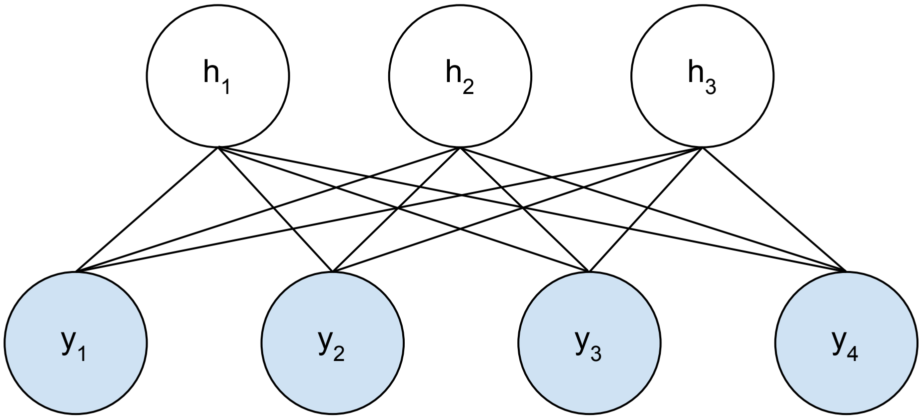


Figure 3.1. An example of a restricted Boltzmann machine (RBM). Visible units y_r are shaded blue and hidden units h_k are unshaded.

3.3.2 Learning

The parameters $\Theta = \{\mathbf{W}, \mathbf{b}, \mathbf{C}\}$ are trained to maximize the log-likelihood of the training data $\{\mathcal{Y}^{(m)}\}_{m=1}^{M}$,

$$L = \max_{\Theta} \sum_{m=1}^{M} \log \left(\sum_{\mathbf{h}} P_{\text{rbm}}(\mathcal{Y}^{(m)}, \mathbf{h}) \right),$$

using stochastic gradient descent. The gradient for the parameters are defined as

$$\frac{\partial L}{\partial b_k} = \left(\frac{1}{M} \sum_{m=1}^{M} P(h_k | y^{(m)}) \right) - P(h_k) \tag{3.9}$$

$$\frac{\partial L}{\partial c_{rl}} = \left(\frac{1}{M} \sum_{m=1}^{M} y_{rl}^{(m)} \right) - P(y_{rl}) \tag{3.10}$$

$$\frac{\partial L}{\partial W_{rlk}} = \left(\frac{1}{M} \sum_{m=1}^{M} y_{rl}^{(m)} P(h_k | y^{(m)}) \right) - P(y_{rl}, h_k) \tag{3.11}$$

The negative components in these gradient are intractable to compute, but they can be approximated using contrastive divergence [37].

3.3.3 Other RBM-based Models

There are several related models that are based on "stacking" RBMs at multiple layers such as the deep belief net (DBN) [39] and the deep Boltzmann machine (DBM) [74]. In deep models, the hidden units at a lower layer act as visible units for the higher layer. Like the RBM, these deep models can learn to generate inputs such as images, but can learn more high-level, complicated dependencies of the input not possible on a 1-layer model like the RBM. It is possible that our work may benefit from the use of a deep model instead of an RBM but training deep models is known to be difficult due to the large number of hyperparameters that need to be carefully determined.

Another model based on the RBM is the conditional restricted Boltzmann machine (CRBM) [86], which is a temporal extension of the RBM. The CRBM has been used to successfully model human motions from motion-capture data, and can generate novel, realistic motion patterns. We use a slightly modified version of the CRBM for use in the semantic labeling of face videos. The CRBM and its usage in our model will be presented in more detail in Chapter 5.

3.4 Discussion

This chapter reviewed the CRF and RBM models which serve as the components used in our later models. The CRF serves as a good baseline for the task of image labeling because it can model local edge interactions between neighboring pixels (or superpixels). However, one limitation of the CRF is that it lacks a global model for label shape. In some cases, the CRF produces labelings that do not *look like* realistic labelings. On the other hand, the RBM can learn realistic models of object shape, as shown by the learned filters and generated samples from Chapter 2. The rest of this thesis presents models that combines these two components to address the limitation of the CRF for labeling tasks. That is, we present models that incorporate the RBM as a global shape prior into the framework of the CRF. This hybrid model is presented in detail in Chapter 4 for images and Chapter 5 for videos.

CHAPTER 4

GLOC

This chapter presents the GLOC model, a strong model for semantic image labeling, which incorporates both local consistency (adjacent nodes that are similar in appearance should have the same label) and global consistency (the overall label shape should look realistic). The GLOC model builds on the CRF and RBM components covered in Chapter 3.

4.1 Introduction

The task of semantic labeling in images is an important problem in computer vision. Labeling semantic regions in an image allows us to better understand the scene itself as well as properties of the objects in the scene, such as their parts, location, and context. This knowledge may then be useful for applications such as object detection or activity recognition.

Huang et al. [45] identified the potential role of semantic labeling in face recognition, noting that a variety of high-level features, such as pose, hair length, and gender can often be inferred (by people) from the labeling of a face image into hair, skin and background regions. This chapter addresses the problem of labeling face regions with hair, skin, and background labels as an intermediate step in modeling face structure. The semantic labeling of face regions may be useful for applications such as recognition, surveillance and retrieval.

As mentioned earlier in Section 1.1, one common approach to semantic labeling is to use a conditional random field (CRF) [52, 80]. Typically, image pixels correspond

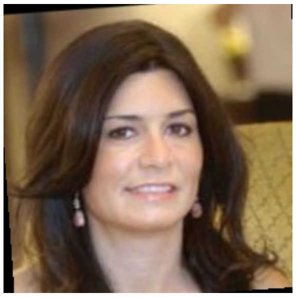
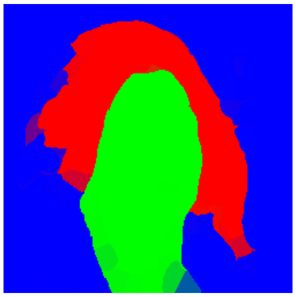
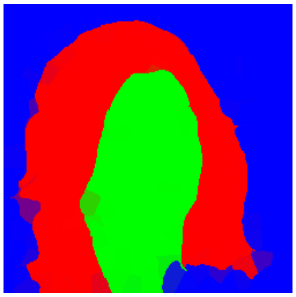
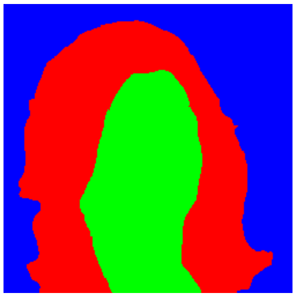
Aligned Image	**CRF**	**GLOC**	**Ground Truth**
			

Table 4.1. An example image and the resulting labeling for Hair/Skin/Background regions using the CRF, GLOC models, and the ground truth labeling. **Red** represents hair, **green** represents skin, and **blue** represents background. The CRF model does not incorporate global label shape and in many cases, such as the given image, the resulting labeling does not *look like* a realistic labeling. The GLOC model which does incorporate global label shape results in a more realistic labeling as compared to the ground truth.

to nodes in a lattice structured graph. Alternatively, mid-level pixel groupings such as superpixels are used as the nodes in a graph. In this case, edges in the graph are placed between adjacent superpixels (i.e. superpixels that share a common boundary). The CRF incorporates edge potentials which frequently help to smooth label boundaries and generally results in a better labeling than a Logistic Regression (LR) model which does not use edge potentials. However, because the CRF typically does not model the global label shape, it can some sometimes produce an unrealistic labeling. As an example, Table 4.1 shows the labeling results for the CRF model and the ground truth labeling, as well as the labeling from the proposed GLOC model. In the table, the CRF labeling (while spatially smooth) simply does not *look* like a realistic labeling.

In many cases, a *global* shape constraint can be used to filter out unrealistic label configurations. It has been shown that restricted Boltzmann machines (RBMs) [82] and their extension to deeper architectures such as deep Boltzmann machines (DBMs) [74], can be used to build effective generative models of object shape (more detail covered in Chapter 2). In particular, the shape Boltzmann machine (SBM) [20] showed impressive performance in generating novel but realistic object shapes while capturing both local and global elements of shape. Motivated by these examples, this chapter presents the *GLOC* (GLObal and LOCal) model, a strong model for image labeling problems, that combines the desirable properties of the CRF (that enforces *local consistency* between adjacent nodes) and the RBM (that models the *global shape prior* of the object).

The GLOC model is evaluated on the face labeling task using the Labeled Faces in the Wild (LFW) [46] data set. As shown later in Section 4.4, the GLOC model brings significant improvements in labeling accuracy over baseline methods, such as the CRF. These gains in numerical accuracy have a significant visual impact on the resulting labeling, often fixing errors that are small but obvious to any observer. Section 4.5 discusses how the hidden units in the GLOC model can be interpreted as face attributes, such as whether an individual has long hair or a beard, or faces to the left or right. These attributes may be useful in retrieving face images with similar structure and properties.

The main contributions in this chapter are as follows:

- The GLOC model, a strong model for face labeling tasks, that combines the CRF and the RBM to achieve both local and global consistency.
- Efficient inference and training algorithms for the GLOC model.
- Significant improvements over the state-of-the-art in face labeling accuracy on subsets of the LFW data set.
- GLOC learns face attributes automatically without attribute labels.

The code [1] and labeled data [2] used in this chapter are publicly available.

4.2 Related Work

As covered earlier in Chapter 1.1, several authors have built systems for segmenting hair, skin, and other face parts [90, 89, 77, 55, 101, 45]. In addition, there has been work in incorporating higher order potentials into CRFs such as the P^N model by Kohli et al. [49]. This model relies on multiple oversegmentations of an image and then incorporates this information into higher order potentials. However, this model does not use global potentials or shape information as we do in our higher potential. Objcut [50] is a system that augments the local information from a CRF with global shape information using pictorial structures (PS) [26, 23]. In comparison, our approaches uses the RBM as a shape prior instead of the PS.

As mentioned in Chapter 2, there have been several related works on using RBMs (or their deeper extensions) for labeling. He et al. [36] proposed multiscale CRFs to model both local and global label features using RBMs. Eslami and Williams [21] proposed a generative model by combining the shape Boltzmann machine (SBM) [20] with an appearance model for parts-based object segmentation. Our model is similar at a high-level to these models in that we use RBMs for object shape modeling to solve image labeling problems. However, there are significant technical differences that distinguish our model from others. First, our model has an edge potential that enforces local consistency between adjacent superpixel labels. Second, we define our model on the superpixel graph using a virtual pooling technique, which is computationally much more efficient. Third, our model is discriminative and can use richer image features than [21] which used a simple pixel-level appearance model (based on RGB pixel values).

Recently, Yujia et al. [60] proposed a similar model to our GLOC model in which they also incorporates a RBM and CRF for image labeling. Both their work and our work were published simultaneously and we were unaware of their work at the time. The main differences between our work and their work are (1) they base their model directly on pixels whereas we use superpixels (which necessitates the need for a virtual pooling layer to map between the fixed grid of the RBM and the superpixels) and (2) they modeled binary label shapes whereas our work focuses on multinomial face label shapes.

4.3 The GLOC Model

The GLOC model incorporates a global label shape prior in the form of the restricted Boltzmann machine (RBM). The RBM is effective at capturing global shape structure through the hidden units and so GLOC *combines* the local modeling from the CRF and the global modeling provided by the RBM. To build a strong model for image labeling, both local consistency (adjacent nodes that are similar in appearance should have the same label) and global consistency (the overall shape of the object should look realistic) are desirable. By "realistic", we mean that the resulting label shapes produced by our model should appear similar to the label shapes in the training data.

We follow the notation for variables introduced earlier in Chapter 3. The conditional likelihood of the labels $\mathcal{Y}$ given the superpixel features $\mathcal{X}$ is defined as follows:

$$P_{\text{gloc}}(\mathcal{Y}|\mathcal{X}) \propto \sum_{\mathbf{h}} \exp\left(-E_{\text{gloc}}(\mathcal{Y}, \mathcal{X}, \mathbf{h})\right), \tag{4.1}$$

$$E_{\text{gloc}}\left(\mathcal{Y}, \mathcal{X}, \mathbf{h}\right) = E_{\text{crf}}\left(\mathcal{Y}, \mathcal{X}\right) + E_{\text{rbm}}\left(\mathcal{Y}, \mathbf{h}\right). \tag{4.2}$$

As described in Equation (4.2), the energy function is a combination of the CRF and RBM energy functions. These energy functions were described previously in Chapter 3 and are reproduced here for convenience.

$$E_{\text{crf}}(\mathcal{Y}, \mathcal{X}) = E_{\text{node}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}\right) + E_{\text{edge}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}\right), \tag{4.3}$$

$$E_{\text{node}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}\right) = -\sum_{s\in\mathcal{V}} \sum_{l=1}^{L} \sum_{d=1}^{D_n} y_{sl}\Gamma_{ld}x_{sd}, \tag{4.4}$$

$$E_{\text{edge}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}\right) = -\sum_{(i,j)\in\mathcal{E}} \sum_{l,l'=1}^{L} \sum_{e=1}^{D_e} y_{il}y_{jl'}\Psi_{ll'e}x_{ije}, \tag{4.5}$$

where $\Gamma \in \mathbb{R}^{L \times D_n}$ represent the node weights and $\Psi \in \mathbb{R}^{L \times L \times D_e}$ is a 3D tensor for the edge weights. For additional details, please refer to Chapter 3. In addition the RBM energy is defined as

$$E_{\text{rbm}}(\mathcal{Y}, \mathbf{h}) = -\sum_{r=1}^{R^2}\sum_{l=1}^{L}\sum_{k=1}^{K} y_{rl} W_{rlk} h_k - \sum_{k=1}^{K} b_k h_k - \sum_{r=1}^{R^2}\sum_{l=1}^{L} c_{rl} y_{rl}, \tag{4.6}$$

where $\mathbf{W} \in \mathbb{R}^{R^2 \times L \times K}$ is a 3D tensor specifying the connection weights between visible and hidden units, b_k is the hidden bias, and c_{rl} is the visible bias.

We now describe how to connect the CRF and RBM components. First note that because the number of superpixels can vary for different images, the RBM energy function in Equation (4.6) requires nontrivial modifications in order to be used with a CRF. That is, we cannot simply connect label (visible) nodes defined over superpixels to hidden nodes as in Equation (4.6) because (1) the RBM is defined over a fixed number of visible nodes and (2) the number of superpixels and their underlying graph structure can vary across images.

4.3.1 Virtual Pooling Layer

To resolve this issue, a *virtual, fixed-sized* pooling layer is used to map between the label and the hidden layers, where each superpixel label node is mapped into the *virtual* visible nodes of the $R \times R$ square grid, where R is the dimension of the grid. This pooling is shown in Figure 4.1, where the top two layers can be thought of as an RBM with the visible nodes $\bar{\mathbf{y}}_r$ representing a surrogate (i.e., pooling) for the labels $\mathbf{y}_s$ that overlap with the grid bin r. Specifically, we define the energy function between the label nodes and the hidden nodes for an image I as follows:

$$E_{\text{rbm}}(\mathcal{Y}, \mathbf{h}) = -\sum_{r=1}^{R^2}\sum_{l=1}^{L}\sum_{k=1}^{K} \bar{y}_{rl} W_{rlk} h_k - \sum_{k=1}^{K} b_k h_k - \sum_{r=1}^{R^2}\sum_{l=1}^{L} c_{rl} \bar{y}_{rl}. \tag{4.7}$$

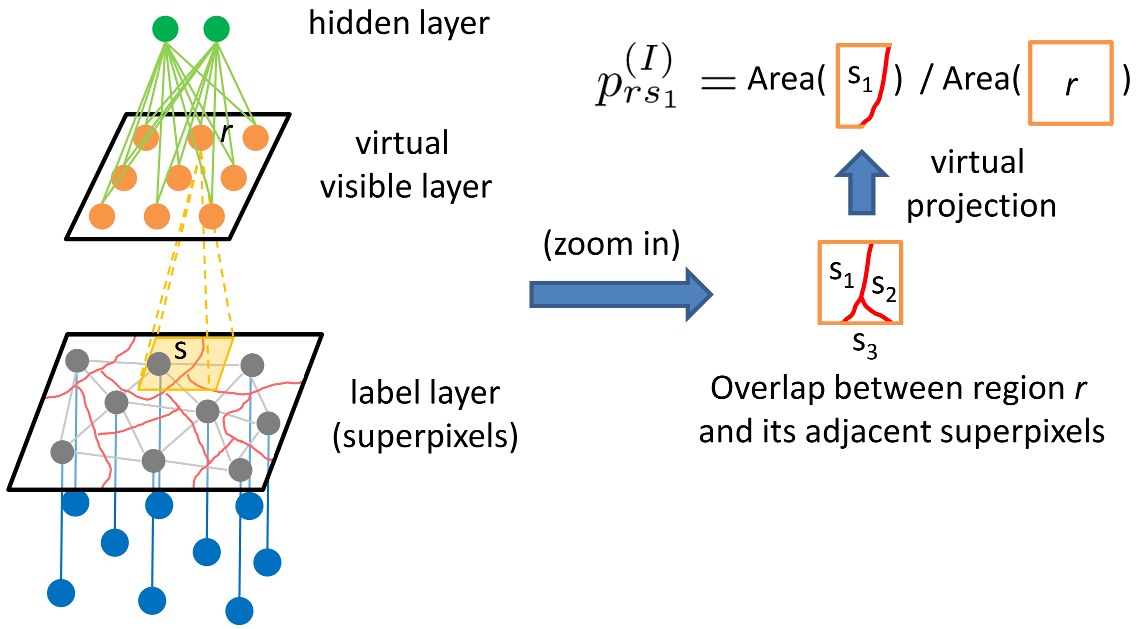

hidden layer
virtual visible layer
r
s
label layer (superpixels)
(zoom in)
$p_{rs_1}^{(I)}$ = Area(s1) / Area(r)
virtual projection
s1
s2
s3
Overlap between region r and its adjacent superpixels

Figure 4.1. The GLOC model. The top two layers can be thought of as an RBM with the (virtual) visible nodes $\bar{\mathbf{y}}_r$ and the hidden nodes h_k. To define the RBM over a fixed-size visible node grid, we use an image-specific "projection matrix" $\{p_{rs}^{(I)}\}$ that transfers (top-down and bottom-up) information between the label layer and the virtual grid of the RBM's visible layer. See text for details.

Recall that $\mathbf{W} \in \mathbb{R}^{R^2 \times L \times K}$ is a weight matrix that specifies the weight connections between virtual visible nodes and hidden nodes, and b_k and c_{rl} are the hidden and visible node biases, respectively. The virtual visible nodes $\bar{y}_{rl} = \sum_{s=1}^{S} p_{rs} y_{sl}$ are deterministically mapped from the superpixel label nodes using the projection matrix $\{p_{rs}\}$ that determines the contribution of label nodes to each node of the grid. The projection matrix is defined as follows:[1]

$$p_{rs} = \frac{Area(Region(s) \cap Region(r))}{Area(Region(r))},$$

[1] The projection matrix $\{p_{rs}\}$ is a sparse, non-negative matrix of dimension $R^2 \times S$. Note that the projection matrix is specific to each image since it depends on the structure of the superpixel graph.

where *Region*(s) and *Region*(r) denote sets of pixels corresponding to superpixel s and grid region r, respectively. Due to the deterministic connection, the pooling layer is actually a *virtual* layer that only exists to map between the superpixel nodes and the hidden nodes. The GLOC model can also be viewed as having a set of grid-structured nodes that performs average pooling over the adjacent superpixel nodes.

4.3.2 Spatial CRF

As mentioned previously in Chapter 3, the CRF can be modified by using location-specific parameters. Specifically, when the object in the image is aligned, we can learn a spatially dependent set of weights that are specific to a cell in an $N \times N$ grid. (Note that this grid can be a different size than the $R \times R$ grid used by the RBM.) We learn a separate set of node weights for each cell in a grid, but the edge weights are kept globally stationary. Using a similar projection technique to that described in Section 4.3.1, the node energy can be defined as

$$E_{\text{node}}(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}) = -\sum_{s \in \mathcal{V}} \sum_{l=1}^{L} y_{sl} \sum_{n=1}^{N^2} p_{sn} \sum_{d=1}^{D_n} \Gamma_{ndl} x_{sd}, \tag{4.8}$$

where $\Gamma \in \mathbb{R}^{N^2 \times D \times L}$ is a 3D tensor specifying the connection weights between the superpixel node features and labels at each spatial location. In this energy function, we define a different projection matrix $\{p_{sn}\}$ which specifies the mapping from the $N \times N$ virtual grid to superpixel label nodes.[2] Note the similarity to the CRF node energy from Equation (4.4). The only difference is the addition of the projection matrix p.

In practice, this spatial CRF tends to perform better than the CRF (as verified by the results in Table 4.2). To demonstrate the utility of localizing the weights for

[2] Note that the projection matrices used in the RBM and spatial CRF are different in that $\{p_{rs}\}$ used in the RBM describes a projection from superpixel to grid ($\sum_{s=1}^{S} p_{rs} = 1$), whereas $\{p_{sn}\}$ used in the spatial CRF describes a mapping from a grid to superpixel ($\sum_{n=1}^{N^2} p_{sn} = 1$).

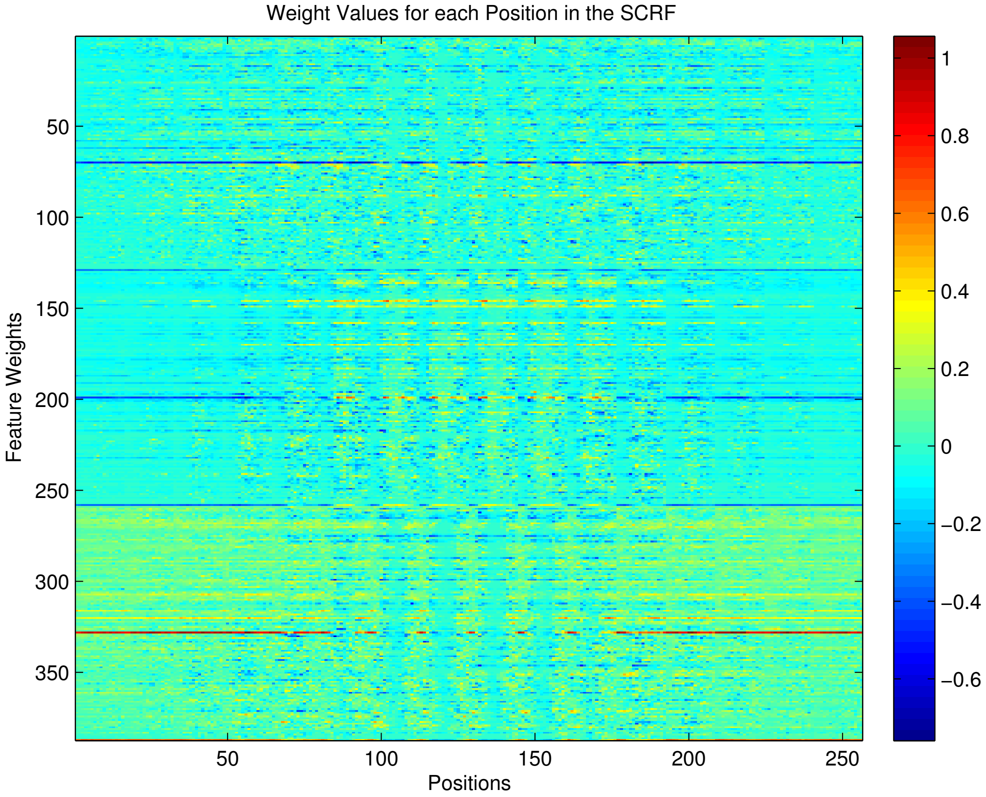


Figure 4.2. SCRF Weights. The dimensions in the SCRF are 128 × 3 × 256, for 128 features, 3 labels, and 256 positions (16 × 16 grid). The figure shows a matrix of dimensions 384 × 256. Each column contains the localized weights for a particular position within the 16 × 16 grid. There is a significant amount of variability between the columns.

each position, the learned node weights Γ for the SCRF are shown in Figure 4.2. The dimensions for the node weights in the SCRF are 128 × 3 × 256, for 128 features, 3 labels, and 256 positions (using a 16 × 16 grid). Recall that the original CRF had dimensions of size 192 × 3 but that 64 position features were removed in the SCRF, in favor of learning location-specific weights. Figure 4.2 shows a matrix of dimensions 384 × 256. Each column shows the localized node weights for a particular position within the 16 × 16 grid. Notice that there is a significant amount of variability between the columns. As before with the CRF, the weights for the SCRF are learned using

Algorithm 3 Mean-Field Inference for GLOC model

```
Initialize $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$ as follows:
    $\mu_{sl}^{(0)} = \frac{\exp\left(f_{sl}^{\text{node}}\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}}\right)}$
    $\gamma_k^{(0)} = sigmoid\left(\sum_{r,l}\left(\sum_s p_{rs}\mu_{sl}^{(0)}\right) W_{rlk} + b_k\right)$
  where
    $f_{sl}^{\text{node}}\left(\mathcal{X}_{\mathcal{V}}, \{p_{sn}\}, \Gamma\right) = \sum_{n,d} p_{sn} x_{sd} \Gamma_{ndl}$
for i=0:$MaxIter$ (or until convergence) do
    update $\boldsymbol{\mu}^{(i+1)}$ as follows: $\mu_{sl}^{(i+1)} =$
        $\frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl}^{\text{rbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl'}^{\text{rbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}$
      where
        $f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}; \mathcal{X}_{\mathcal{E}}, \mathcal{E}, \Psi\right) = \sum_{j:(s,j)\in\mathcal{E}} \sum_{l',e} \mu_{jl'} \Psi_{ll'e} x_{sje}$
        $f_{sl}^{\text{rbm}}\left(\boldsymbol{\gamma}; \{p_{rs}\}, \mathbf{W}, \mathbf{C}\right) = \sum_{r,k} p_{rs}\left(W_{rlk}\gamma_k + c_{rl}\right)$
    update $\boldsymbol{\gamma}^{(i+1)}$ as follows:
        $\gamma_k^{(i+1)} = sigmoid\left(\sum_{r,l}\left(\sum_s p_{rs}\mu_{sl}^{(i+1)}\right) W_{rlk} + b_k\right)$
end for
```

the minFunc[3] software package.

4.3.3 Inference

Since the joint inference of superpixel labels and hidden nodes is intractable, an approximate inference approach is necessary. In this thesis, a mean-field approximation is used (described previously in Chapter 3). The mean-field inference steps are described in Algorithm 3.

[3] http://www.di.ens.fr/~mschmidt/Software/minFunc.html

The variational parameters $\boldsymbol{\mu}_{sl}^{(i)}$ and $\boldsymbol{\gamma}_k^{(i)}$ correspond to current estimates of the node labels and the hidden units, respectively. Just as in the mean-field approximation for the SCRF, the parameters $\boldsymbol{\mu}_{sl}^{(0)}$ are initialized to the logistic regression guess in Step 1. Similarly, the parameters $\boldsymbol{\gamma}_k^{(0)}$ are initialized to the posterior of the hidden units, given the guesses for the label nodes $\boldsymbol{\mu}_{sl}^{(0)}$. The algorithm then iterates until the maximum number of iterations is reached (denoted by $MaxIter$) or when the variational parameters no longer change after updates.

4.3.4 Learning

The model parameters $\{\mathbf{W}, \mathbf{b}, \mathbf{C}, \Gamma, \Psi\}$ are trained to maximize the conditional log-likelihood. The gradient of the full model is given by

$$\nabla_\theta \log p\left(Y|X\right) = \mathbb{E}_{p(H|Y,X)}\left[-\nabla_\theta E\right] - \mathbb{E}_{p(H,Y|X)}\left[-\nabla_\theta E\right],$$

which is a difference between the expectations of the data and model terms.

In practice, however, it is beneficial to provide a proper initialization (or *pretrain*) to those parameters. An overview of the training procedure is shown in Algorithm 4. The GLOC model can be trained to either maximize the conditional log-likelihood using contrastive divergence (CD) or minimize the generalized perceptron loss [54] using CD-PercLoss [65]. In fact, Mnih et al. [65] suggested that CD-PercLoss would be a better choice for structured output prediction problems since it directly penalizes the model for wrong predictions during training. We empirically observed that CD-PercLoss performed slightly better than CD for our labeling task.

4.3.5 Piecewise Model

One drawback of joint training the GLOC model is that it is necessary to carefully set hyperparameters, such as regularization parameters and learning rates. A simple alternative is to learn the RBM and CRF components separately and combine them

Algorithm 4 Training GLOC model

```
1: Pretrain $\{\Gamma, \Psi\}$ to maximize the conditional log-likelihood of the *spatial CRF* model (See Equations (4.3), and (4.8)).
2: Pretrain $\Theta = \{\mathbf{W}, \mathbf{b}, \mathbf{C}\}$ to maximize the conditional log-likelihood $\log \sum_{\mathbf{h}} P_{\text{crbm}}(\mathcal{Y}, \mathbf{h}|\mathcal{X}_{\mathcal{V}})$ of the GLOC model without edge potentials which is defined as:
       $P(\mathcal{Y}, \mathbf{h}|\mathcal{X}_{\mathcal{V}}) \propto \exp\left(-E_{\text{node}}(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}; \Gamma) - E_{\text{rbm}}(\mathcal{Y}, \mathbf{h}; \Theta)\right)$
3: Train $\{\mathbf{W}, \mathbf{b}, \mathbf{C}, \Gamma, \Psi\}$ to maximize the conditional log-likelihood of the *GLOC* model (See Equation (4.1)).
```

in a piecewise model. A single scalar parameter λ represents the tradeoff between the RBM contribution and the CRF contribution as shown in Equation (4.9). The parameter λ is determined by trying a range of values between [0..1] and choosing the value resulting in the best accuracy on the validation set. In these experiments, $\lambda = 0.2$.

The node update for $\mu_{sl}^{(i+1)}$ in Algorithm 3 is replaced by :

$$\mu_{sl}^{(i+1)} = \frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + \lambda f_{sl}^{\text{rbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + \lambda f_{sl'}^{\text{rbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}. \tag{4.9}$$

The node update μ_{sl} is the same as before in Algorithm 3, but now λ is used to weight the contribution from the RBM. In practice, this piecewise model performs well, but slightly worse than the jointly trained model as shown in Table 4.2.

4.4 Experiments

The proposed GLOC model is evaluated on a task to label face images from the LFW database [46] as hair, skin, or background regions. We use the "funneled" version of LFW, in which images have been coarsely aligned using a congealing-style joint alignment approach [43]. Although some better automatic alignments of these images exist, such as the LFW-a database [98], LFW-a does not contain color information, which is important for our application. A newer alignment algorithm using deep models was introduced recently by Huang et al. [44], but this work was not yet published at the time of these experiments.

The LFW website provides the segmentation of each image into superpixels, which are small, relatively uniform pixel groupings.[4] We provide ground truth for a set of 2927 LFW images by labeling each superpixel as either hair, skin, or background (this data is publicly available [2]). While some superpixels may contain pixels from more than one region, most superpixels are generally "pure" hair, skin, or background. We use a superpixel representation instead of a pixel-level representation mostly for computational efficiency (more discussion can be found in Chapter 3).

The algorithm used to generate superpixels is from Mori et al. [66]. At the time of our experiments, this approach was among the top performing superpixel algorithms whose code was publicly available. Since then, newer methods like SLIC [5] have become popular. However, in their paper, the authors of SLIC compared several superpixel algorithms including the approach by Mori et al. [66] and found that the algorithm by Mori et al. [66] still compares favorably, while being slower and requiring more memory than SLIC.

4.4.1 Features

The set of features is the same as in Huang et al. [45]. For each superpixel the following node features are computed.

- **Color**: Normalized histogram over 64 bins generated by running K-means over pixels in LAB space. Image pixels are first clustered using K-means, using $K = 64$, and each pixel is assigned to its closest centroid. Afterward a normalized histogram is computed using all the pixel assignments within a superpixel.

[4] Available at `http://vis-www.cs.umass.edu/lfw/lfw_funneled_superpixels_fine.tgz`.

- **Texture**: Normalized histogram over 64 textons which are generated according to [63]. To generate textons, we first convolve a set of training images with a filterbank and gather the filter responses. In our experiments, the filterbank consists of 12 filters at varying orientations, and at 3 different scales for a total of 36 filters. Specifically, the 3 sets of filters were of dimensions $19 \times 19, 27 \times 27$, and 39×39. Each pixel in the image now has a corresponding vector of 36 filter responses. These filter responses are then clustered using K-means into bins or *textons* (in our case we cluster into 64 textons). For a new image, the image is convolved with the filterbank to get filter responses for each pixel, and these pixel responses are then assigned to the closest texton. Afterward, a normalized histogram is computed for all the pixel assignments within a superpixel.

- **Position**: Normalized histogram of the proportion of a superpixel that falls within the 8×8 grid overlayed on the image.[5]

The following edge features were computed between adjacent superpixels:

- **Probability of Boundary (Pb)** [64]: Sum of the Pb values between adjacent superpixels.

- **Color**: L2 distance between color histograms for adjacent superpixels.

- **Texture**: Chi-squared distance between texture histograms for adjacent superpixels as computed in [45]

  $$\chi^2 = \frac{1}{2} \sum_{i=1}^{64} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)},$$

  where h_1, h_2 refer to the texture histograms of adjacent superpixels.

[5] Note that the position feature is only used in the CRF.

4.4.2 Evaluation

The labeling performances of several different models are compared in Table 4.2. The labeled examples are split into training, validation, and testing sets that contain 1500, 500, and 927 examples, respectively. The models evaluated are

- Logistic Regression (LR)
- Spatial Logistic Regression (SLR)
- Conditional Random Field (CRF)
- Spatial Conditional Random Field (SCRF)
- SLR + RBM
- SCRF + RBM (GLOC)

These models range from the simple LR model to our GLOC model. The GLOC model was trained using batch gradient descent and model hyperparameters were selected that performed best on the validation set. The hyperparameters used are K=400, R=24, and N=16. All models were trained using LBFGS optimization from the minFunc [3] software package. On a multicore AMD Opteron, average inference time per example was 0.254 seconds for the GLOC model and 0.063 seconds for the spatial CRF.

The following metrics are used for evaluation in Table 4.2:

- **Error reduction**: computed as the following, with respect to the SCRF baseline:

  $$\text{Error Reduction(model)} = \frac{[100 - \text{Accuracy(CRF)}] - [100 - \text{Accuracy(model)}]}{100 - \text{Accuracy(CRF)}}.$$

  Error reduction is shown with respect to the CRF because it is a typical approach used for semantic labeling [45, 80].

Model	Error Reduction	Accuracy	Hair	Skin	BG	Avg
LR	-34.004	90.922	56.800	91.723	95.848	81.457
SLR	-15.092	92.203	62.903	93.363	96.282	84.183
CRF	0	93.226	73.682	93.953	95.961	87.865
SCRF	10.639	93.946	73.895	94.809	96.715	88.473
SLR + RBM	12.393	94.065	75.056	93.583	97.027	88.555
SCRF + RBM = GLOC (PW)	15.534	94.278	74.049	94.098	97.322	88.490
SCRF + RBM = GLOC (Joint)	**25.412**	**94.947**	**78.687**	**94.833**	**97.453**	**90.324**

Table 4.2. Labeling accuracies for each model over superpixels. The columns correspond to: 1) model name, 2) error reduction over the CRF 3) overall superpixel labeling accuracy 4,5,6) category-level superpixel accuracies for Hair/Skin/Background and 7) average category-level accuracy (i.e. the average of columns 4-6). All results are in percentages. The best results for columns 2-7 are shown in **bold.**

- **Overall superpixel accuracy**: the number of superpixels classified correctly divided by the total number of superpixels, across all folds.
- **Category-specific superpixel accuracy**: for each class, the number of superpixels classified correctly divided by the total number of superpixels, across all folds
- **Average category-specific superpixel accuracy**: the average of the category-specific superpixel accuracies.

The results using these different models are shown in Table 4.2. As shown in the table, there is a significant improvement for the GLOC model in superpixel labeling accuracy over the baseline CRF and LR models. While absolute accuracy improvements (necessarily) become small as accuracy approaches 95%, the reduction in errors are substantial. The GLOC model has significant improvements in not just the raw superpixel accuracy (column 3 of Table 4.2), but also the per-category accuracies. For each category, the GLOC model outperformed the other models. In addition, for both the logistic regression (LR) and conditional random field (CRF) models, the spatial version (SLR and SCRF respectively) outperformed the non-spatial version of

		Ground Truth		
		Hair	Skin	Background
Guess	Hair	7.928	0.448	1.097
	Skin	0.586	20.723	0.656
	Background	1.531	0.744	66.286

Table 4.3. Confusion matrix for GLOC model. The majority of the errors mistake the Hair and Background classes.

the model. Table 4.3 shows the confusion matrix for the GLOC model. The majority of the errors mistake the Hair and Background classes. Examples of these errors are shown later in Table 4.6.

By analyzing the performance of the different models in Table 4.2 we can examine the relative contributions of the local and global potentials, as well as spatial versions of the models. Local potentials are provided by the edges in the CRF and global potentials are provided by the RBM. The spatial models (SLR and SCRF) overlay an $N \times N$ grid on top of the image and learn node weights specific to each grid region.

For both the LR and CRF models, the spatial versions (SLR and SCRF, respectively) outperformed the non-spatial versions. That is, SLR outperforms LR by about 1.28% superpixel accuracy and SCRF outperforms CRF by about 0.72% superpixel accuracy. Since SLR outperforms LR, the SLR model is used as the baseline for future comparisons.

Recall that the SLR model has neither local modeling nor global modeling, and that each superpixel is treated independently. Adding local potentials (SCRF) provides a 1.74% improvement in superpixel accuracy. Adding the global potentials (SLR+RBM) without the local potentials provides a 1.86% improvement in superpixel accuracy. Adding both local and global potentials (GLOC) provides a 2.74% improvement in superpixel accuracy over the SLR model.

Furthermore, there are significant qualitative differences in many cases, as illustrated in Tables 4.4 and 4.5. In particular, Table 4.4 shows significant improvement

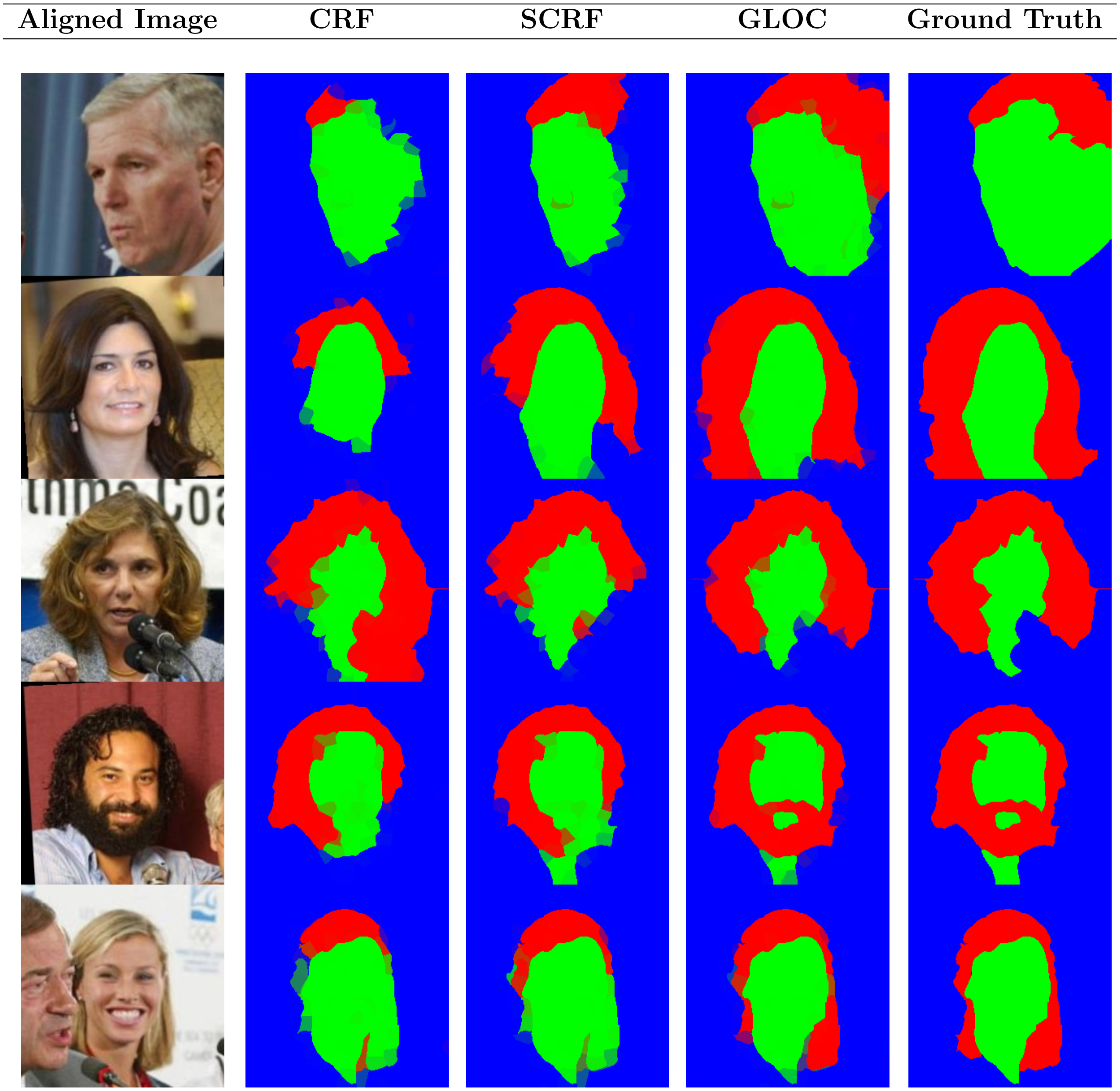


Table 4.4. Large Improvement. Successful labeling results on images from the LFW database. This table shows images in which the GLOC model made relatively large improvements over the baseline SCRF. Note that the SLR+RBM results are not shown here.

of GLOC over the SCRF, and Table 4.5 show more subtle improvements made by the GLOC model. The images contain extremely challenging scenarios such as multiple distractor faces, occlusions, strong highlights, and pose variation. The confidence of the guess (posterior) is represented by color intensity. A confident guess appears as

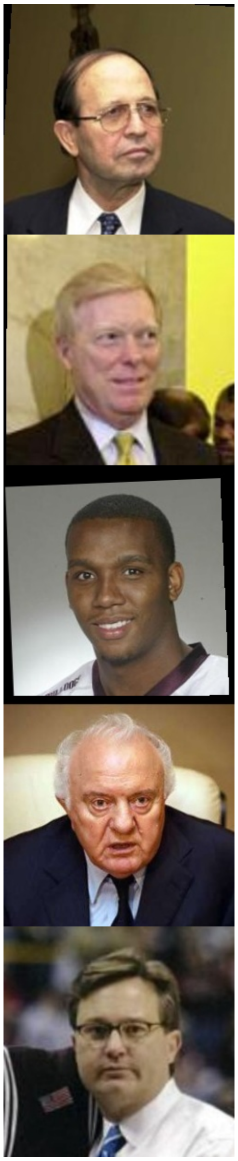
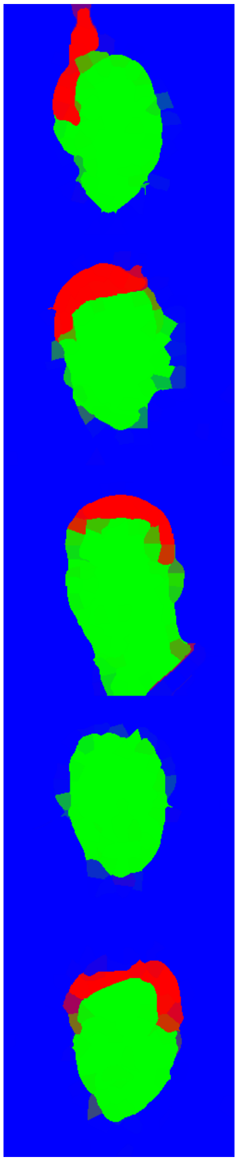
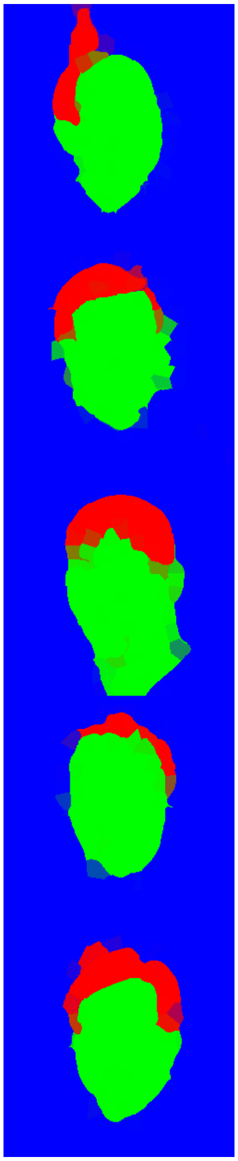
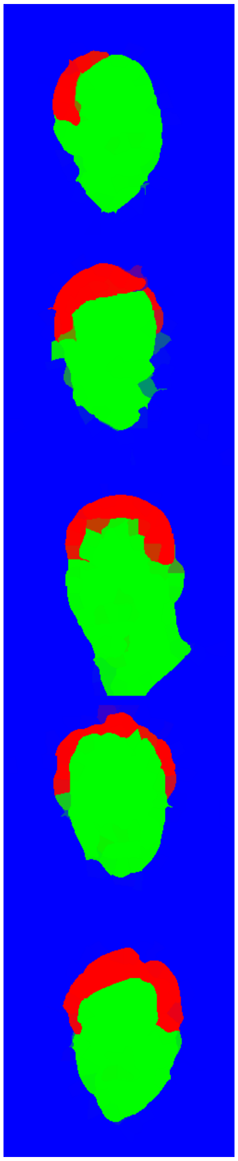
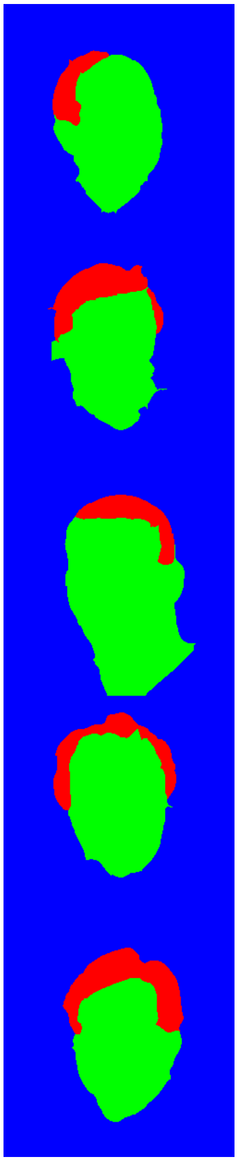
Aligned Image	CRF	SCRF	GLOC	Ground Truth
				

Table 4.5. Subtle Improvement. Successful labeling results on images from the LFW database. This table shows images in which the GLOC model made relatively small, more subtle improvements to the baseline SCRF. Note that the SLR+RBM model results are not shown here.

a strong **red**, **green**, or **blue** color, and a less confident guess appears as a lighter mixture of colors. As we can see, the global shape prior of the GLOC model helps "clean up" the guess made by the SCRF in many cases, resulting in a more confident prediction.

In many cases, the RBM prior encourages a more realistic labeling by either “filling in” or removing parts of the hair or face shape. For example, the woman in the second row in Table 4.4 recovers the left side of her hair and gets a more recognizable hair shape under the GLOC model. Also, the man in the first row in Table 4.5 gets a more realistic looking hair shape by removing the small (incorrect) hair shape on top of his head. This effect may be due to the top-down global prior in the GLOC model, whereas simpler models such as the SCRF do not have this information. In addition, there were cases (such as the woman in the fifth row Table 4.4) where an additional face in close proximity to the centered face may confuse the model. In this case, the CRF and SCRF models make mistakes, but since the GLOC model has a strong shape model, it was able to find a more recognizable labeling of the foreground face.

On the other hand, the GLOC model sometimes makes errors. Typical failure examples are shown in Table 4.6. The model made significant errors in the hair regions (shown quantitatively in the confusion matrix in Table 4.3). Specifically, in the first row, the hair of a nearby face is similar in color to the hair of the foreground face as well as the background, and GLOC incorrectly guesses more hair by emphasizing the hair shape prior, perhaps too strongly. In addition, there are cases in which occlusions cause problems, such as the third row. However, we point out that occlusions are frequently handled correctly by our model (e.g. the microphone in the third row in Table 4.4).

Figure 4.3 shows the difference between the GLOC and SCRF labeling accuracy for all 927 test images, sorted in increasing order. Specifically, for each test case, the difference between the GLOC superpixel accuracy and SCRF superpixel accuracy was computed and then these differences were sorted in increasing order. Overall, there is a net improvement by using GLOC instead of the SCRF. For about 150 test cases, there is no difference between the two models.

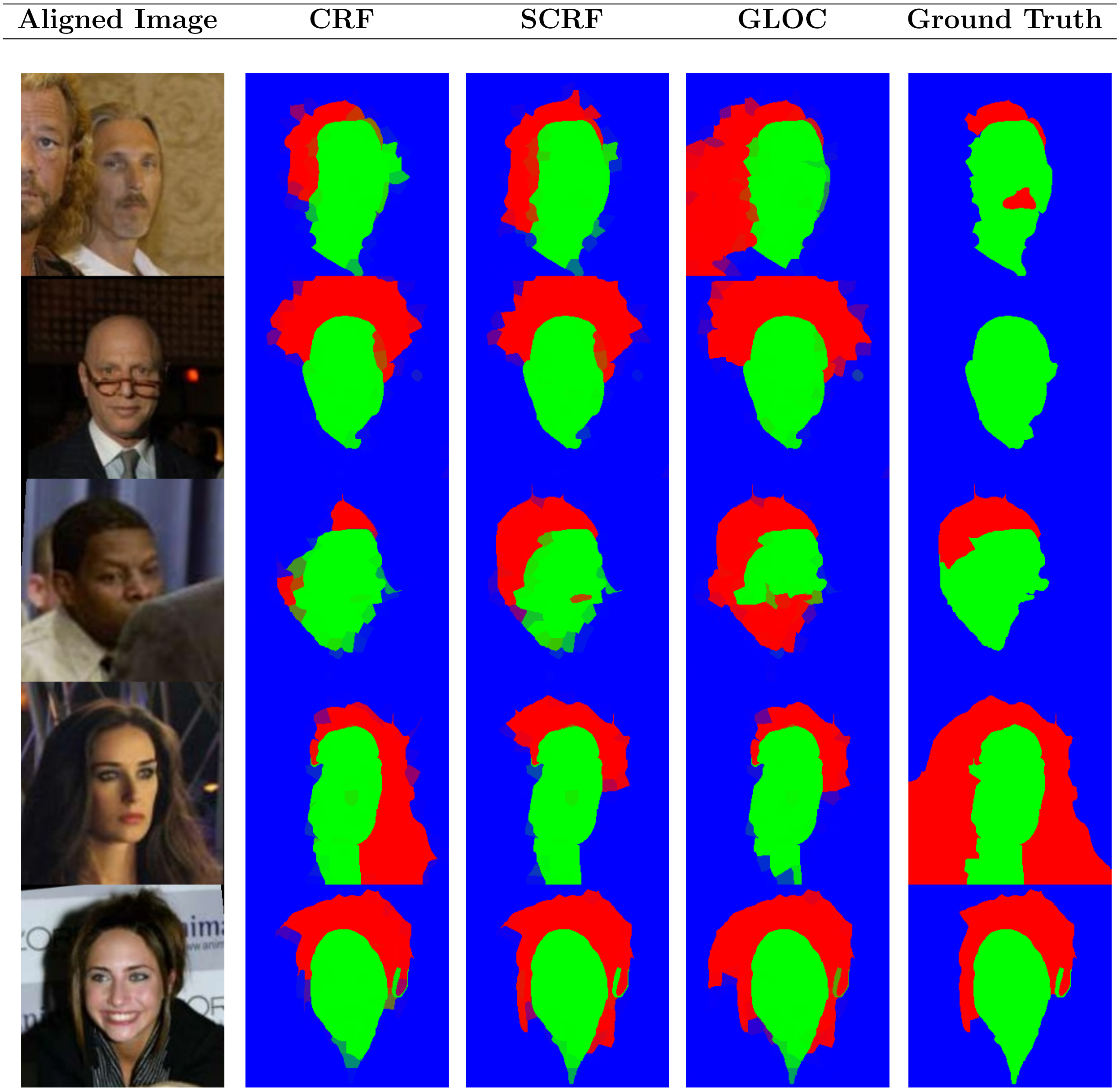


Table 4.6. Typical Failure Cases. This figure shows typical failure cases made by the GLOC model. GLOC makes errors due to factors such as additional faces in close proximity to the centered face, shadows, and occlusions.

4.4.3 Comparison to Prior Work

Wang et al. [90] also did work in the semantic labeling of faces based on templates, as mentioned earlier in Section 4.2. Their database contains 1046 LFW (unfunneled) images whose pixels are manually labeled for four regions (Hair, Skin, Background,

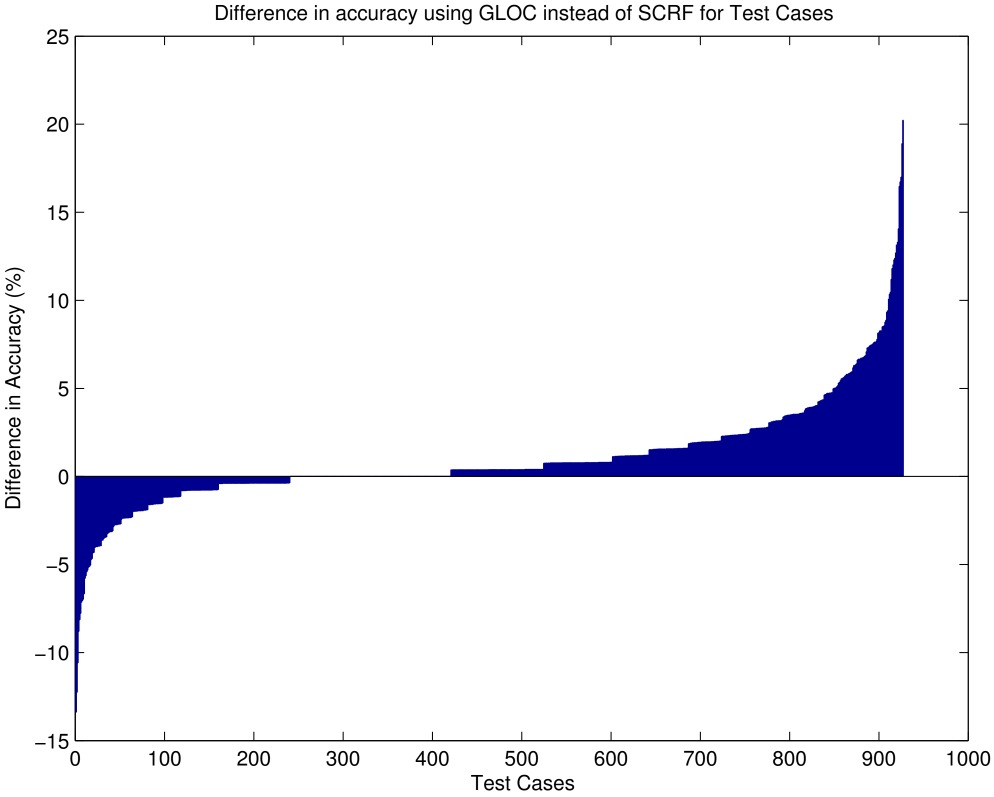


Figure 4.3. This figure shows the difference between the GLOC and SCRF superpixel labeling accuracy for all 927 test images, sorted in increasing order. Overall, there is a net improvement by using GLOC instead of the SCRF.

and Clothing). Since their code is unavailable, we were unable to run their code on our own data, but we were able to run our models on their data. Following their evaluation setup, the data was randomly split in half with one half used for training and the other half for testing. This procedure is repeated five times and the average pixel accuracy is reported as the final result.

Following the approach used for LFW images, we generated a superpixel segmentation for each image (using the method of Mori et al. [66], the same approach used in the GLOC model experiments), features for each image, trained a new GLOC model, and then ran the model to get label guesses for each superpixel. Afterward, the label guesses were mapped back to pixels for evaluation (recall that the ground truth is

		Ground Truth			
		Hair	Skin	Clothing	Background
Guess	Hair	13.922	0.393	0.159	0.416
	Skin	0.300	16.049	0.201	0.123
	Clothing	0.126	0.214	18.191	0.196
	Background	0.419	0.106	0.181	49.004

Table 4.7. Confusion matrix for superpixel mapping.

provided in pixels). Each pixel within a superpixel is assigned to the label guess for the superpixel.

Even with a "perfect" superpixel labeling, this mapping incurs approximately 2.83% error. That is, if we use the ground truth to pick the majority label for each superpixel, and then map this labeling back to pixels, this incurs about a 2.83% pixel-level error. However, even accounting for this mapping error, our approach was still sufficient to obtain a pixel-wise accuracy of 90.7% which improves by 0.7% upon the best reported result of 90.0% from Wang et al. [90].

The confusion matrix is shown in Table 4.7 where the guesses correspond to the superpixel labels that have been mapped into pixels. As before with the confusion matrix for GLOC in Table 4.3, most of the errors mistake the hair and background classes. There is about a 2.83% overall labeling error indicating that even though there is some error incurred by using superpixels, it is not excessive. Therefore, using superpixels may still offer a good tradeoff between computational complexity and this error.

4.5 Attributes and Retrieval

While the labeling accuracies shown in Table 4.2 are a direct evaluation of our models, there is an additional goal in our work: to build models that capture the natural statistical structure in faces. It is not an accident that human languages have words for beards, baldness, and other salient high-level attributes of human face

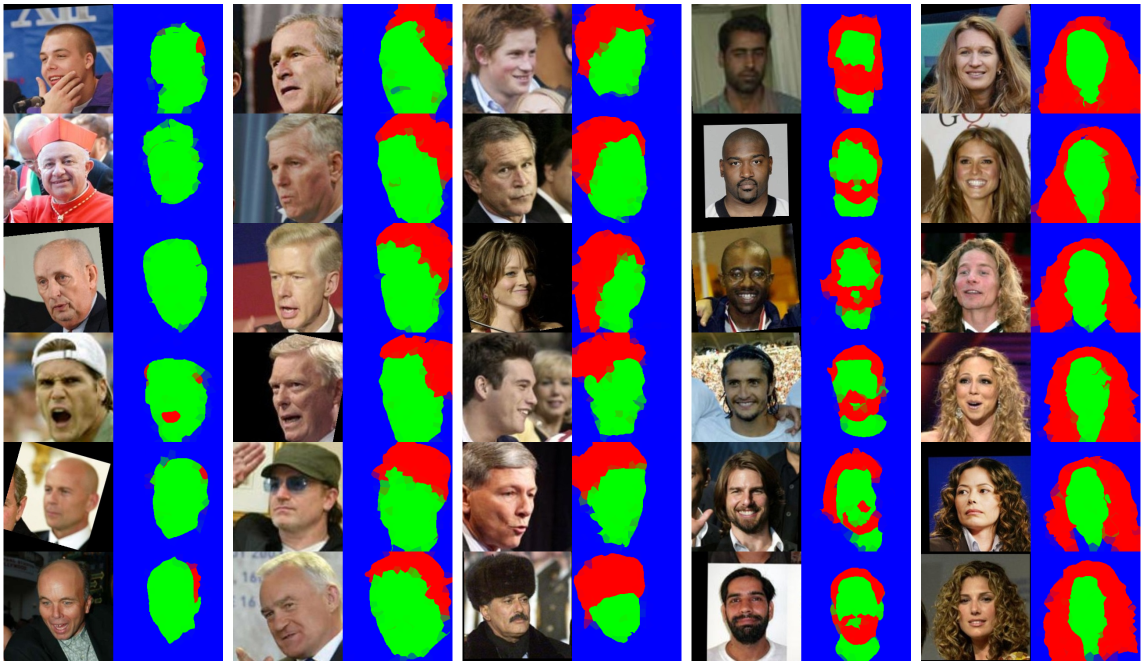


Figure 4.4. This figure shows some of the latent structure automatically learned by the GLOC model. In each column, we retrieve the images from LFW (except images used in training and validation) with the highest activations for each of 5 hidden units, and provide their labeling results. The attributes from left to right can be interpreted as "no hair showing", "looking left", "looking right", "beard/occluded chin", and "big hair". Although the retrieved matches are not perfect, they clearly have semantic, high-level content.

appearance. These attributes represent coherent and repeated structure across the faces we see everyday. Furthermore, these attributes are powerful cues for recognition, as demonstrated by Kumar et al. [51]. One of the most exciting aspects of RBMs and their deeper extensions are that these models can learn latent structure automatically. Recent work has shown that unsupervised learning models can learn meaningful structure without being explicitly trained to do so (e.g., [53, 42, 44]).

In our experiments, we ran the GLOC model on all LFW images other than those used in training and validation, and sorted them based on each hidden unit activation. Each of the five columns in Figure 4.4 shows a set of retrieved images and their guessed labelings for a particular hidden unit. In many cases, the retrieved results for the hidden units form meaningful clusters. These units seem highly correlated with "lack of hair", "looking left", "looking right", "beard or occluded chin", and "big hair". Thus, the learned hidden units may be useful as attribute representations for faces. Figure 4.4 shows the retrieved results for five hidden units with very salient structure. Note that our model uses $K = 200$ hidden units. Many of the resulting matches for the other remaining hidden units are similar to the matches shown in Figure 4.4.

4.6 LFW Verification

Labeled Faces in the Wild (LFW) [46] was designed to study face recognition. It is one of the main benchmarks for the face verification task, which is the task of telling whether two faces are of the same person or not. It is possible that the labelings from the GLOC model may provide useful, additional information for this task. For example, knowing the general shape of the Hair/Skin/Background regions may be useful to a classifier trained for face verification. Many current system do not currently use this face shape information.

Li et al. [56] currently have one of the top-performing models on the image-restricted LFW evaluation (in this setting, additional training data is not allowed). Their reported accuracy on this task is 0.8408 ± 0.0120. They have graciously assisted us by incorporating the GLOC labelings for LFW images into their own features and re-running their models. They have reported a small improvement of about 0.0025 [41]. While this is a small improvement and it is within the error bounds, it may still demonstrate that there is some signal provided by the face labelings. However, additional work is necessary to gain a significant improvement

4.7 Hyperparameter Selection

One of the drawbacks of working with a model of many different types of parameters is how to choose the hyperparameters. Typically, hyperparameters are chosen based on what performs best on the validation set. This section looks in detail at one important hyperparameter, the number of hidden units to use in the RBM. A small number of hidden units may not be enough to capture the desired dependencies in the data, but too many hidden units may begin to overfit the data. It is also desirable to keep the number of parameters in the model from being too large. Figure 4.5 shows the result of varying the number of hidden units in the GLOC model on the validation set, shown at both regular and log scale. As shown in the figure, after about 50 hidden units, performance does not change significantly, as the superpixel validation accuracy settles at about 94.6%.

4.8 Discussion

Face segmentation and labeling is challenging due to the diversity of hair styles, head poses, clothing, occlusions, and other phenomena that are difficult to model, especially in a real world database like LFW. The GLOC model combines the CRF and the RBM to model both local and global structure in face labelings. GLOC has consistently reduced the error in face labeling over baseline models which lack global shape priors. In addition, we have shown that the hidden units in our model can be interpreted as face attributes, which were learned without any attribute-level supervision.

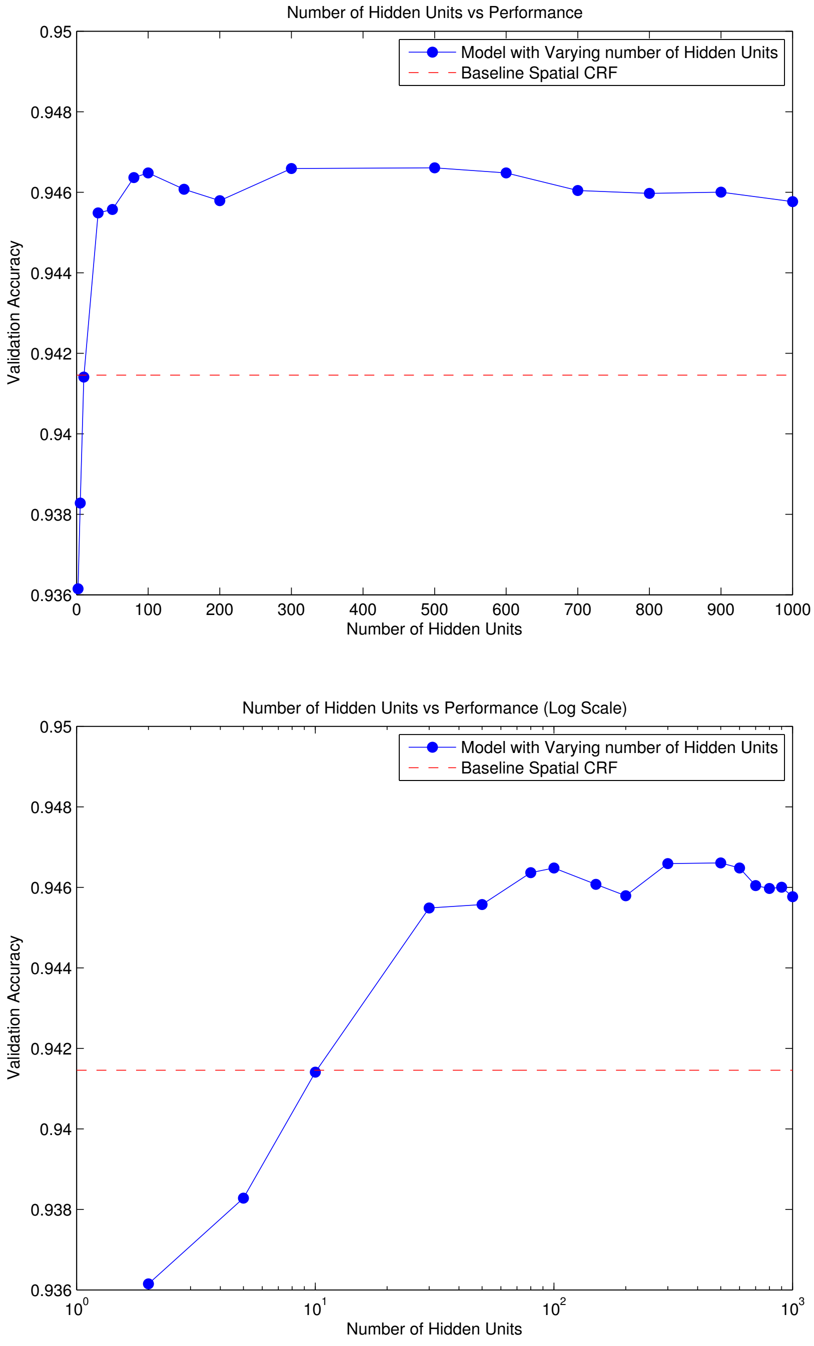


Figure 4.5. These figures show validation accuracy with a variable number of hidden units. The bottom figure is a log scale version of the top figure. After about 50 hidden units, performance does not change significantly.

CHAPTER 5

SHAPE-TIME RANDOM FIELD

This chapter presents the Shape-Time Random Field (STRF) model, a strong model for semantic video labeling. It can be considered a temporal extension of the GLOC model presented in the previous chapter. STRF incorporates not only local consistency (adjacent nodes are likely to have similar labels) and global consistency (the overall shape of the object should look realistic) as in the previous GLOC model, but it also incorporates temporal consistency. The STRF model builds on the CRF and RBM components covered in Chapter 3.

5.1 Introduction

The task of semantic labeling in video is interesting to study because there is typically more information available in a video of an object than a static image of an object. For example, we can track the motion of an object in video and learn properties about the object, such as the way the object moves and interacts with its environment, which is more difficult to infer from a static image. In addition, there are many videos publicly available on sites such as YouTube, which makes analysis in videos increasingly useful.

We again focus on performing a semantic labeling of hair, skin, and background regions but from videos and not static images. An example clip from a video and its corresponding labeling is shown in Table 5.1. Such a labeling may be useful for other tasks such as surveillance and face recognition. The previous chapter presented the GLOC model, which incorporated a label prior (in the form of a restricted Boltzmann

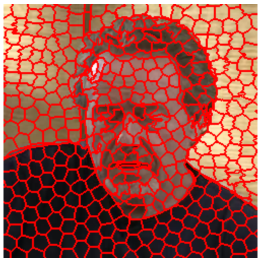
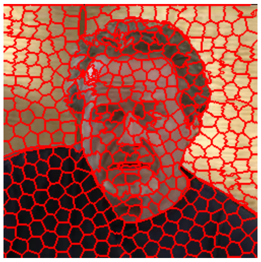
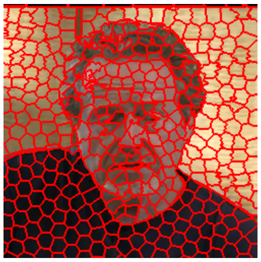
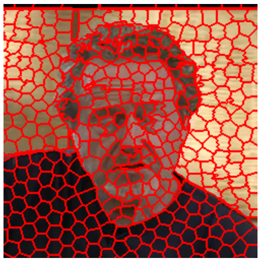
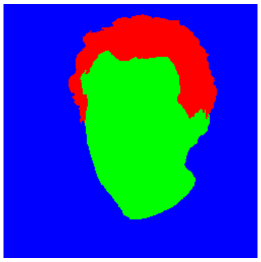
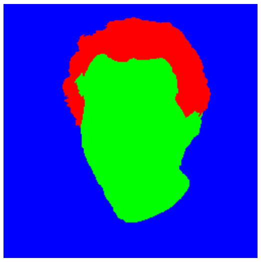
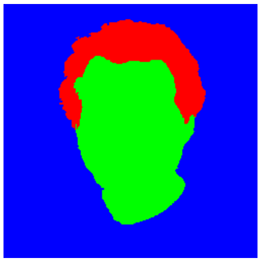
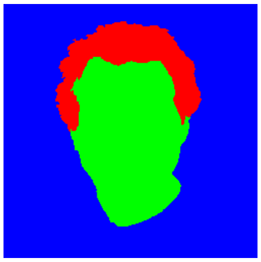
	t	t+2	t+4	t+6
YFDB				
Superpixel				
Ground Truth				

Table 5.1. The first row shows a clip from the YFDB. The second rows shows the temporal superpixel segmentation and the last row shows ground truth. **Red** represents hair, **green** represents skin, and **blue** represents background.

machine (RBM) [82]) into the framework of a conditional random field (CRF) [52] model and showed that it improved labeling accuracy over a baseline CRF. This model is appropriate for images since it accounts for global and local dependencies, but it does not account for temporal dependencies present in video.

While an RBM can be used to model temporal dependencies, it may be more efficient to use a conditional restricted Boltzmann machine (CRBM) [86]. The CRBM is an extension of the RBM to account for temporal dependencies by looking at a window of previous frames in a video. It was used to model motion capture data and was able to generate novel motions [86]. An important distinction is that we use the CRBM for improving classification accuracy. In our model, we use the CRBM to provide a *dynamic bias* for the current frame by conditioning on the labels of the previous frames and thereby better inform the label shape of the current frame. For example, if the previous frames show a person with their head posed to the left, then it is reasonable to assume that in the current frame, the person may still have their head posed to the left.

We incorporate the CRBM as a prior which models both temporal and object shape dependencies into the framework of a CRF, which can provide local modeling. This combined model is referred to as the Shape-Time Random Field (STRF) because it models both shape and temporal dependencies within a random field framework. As we show in our results, STRF outperforms competitive baselines for the task of labeling hair, skin, and background regions in face videos. In addition, we introduce a new database of labeled hair, skin, and background regions (original videos from the YouTube Faces DB[97]). Both the code and labeled data will be made publicly available upon publication.

5.2 Related Work

Conditional random fields (CRFs) [52] have been used widely in image labeling tasks [35, 36, 80, 6, 8] where nodes are defined over either a pixel or superpixel grid, and edges are defined over neighboring pixels or adjacent superpixels. One straightforward way to extend these models to label videos is to define temporal potentials between frames within a small neighborhood [92, 27].

The restricted Boltzmann machine (RBM) [82] and related deep models (such as the deep Boltzmann machine (DBM) [74]) have demonstrated impressive generative abilities for learning object shape. Salakhutdinov et al. [74] trained a DBM to learn and generate novel digits and images of small toys. Recently, Eslami et al. [20] introduced the Shape Boltzmann Machine (SBM) as a strong model of object shape, in the form of a modified DBM. The SBM was shown to have good generative performance in modeling simple, binary object shapes. The SBM was later extended to perform classification within a generative model [21].

Because we are interested in modeling the shape of an object over time, we use the conditional restricted Boltzmann machine (CRBM) introduced by Taylor et al. [86]. The CRBM is an extension of the RBM with additional connections to a history of previous frames. They used the CRBM to learn different motion styles from motion-captured data, and successfully generated novel, realistic motions. In this thesis, the CRBM is used not to generate realistic data, but to model temporal dependencies in video and help improve labeling performance.

5.3 Models

This section presents the components of the Shape-Time Random Field (STRF) model which include the CRF and RBM, previously covered in Chapter 3.

Notation. Let us review the mathematical notation used to describe the models. The notation is very similar to the notation used in previous chapters, except we now account for frames in a video and not just individual static images.

- A video v consists of $F^{(v)}$ frames, where $F^{(v)}$ can vary over different videos. Let each frame in video v be denoted as $v^{(t)}$ where $t \in \{1 \cdots F^{(v)}\}$.

- A video frame $v^{(t)}$ is pre-segmented into $S^{(v,t)}$ superpixels, where $S^{(v,t)}$ can vary over different frames. The superpixels represent the nodes in the graph for video v at time t.

- Let $\mathcal{G}^{(v,t)} = \{\mathcal{V}^{(v,t)}, \mathcal{E}^{(v,t)}\}$ denote the nodes and edges for the undirected graph of frame t in video v.

- Let $\mathcal{V}^{(v,t)} = \{1, \cdots, S^{(v,t)}\}$ denote the set of superpixel nodes for frame t in video v.

- Let $\mathcal{E}^{(v,t)} = \{(i,j) : i,j \in \mathcal{V}^{(v,t)}$

  and i, j are adjacent superpixels in frame t in video $v\}$.

- Let $\mathcal{X}^{(v,t)} = \{\mathcal{X}_{\mathcal{V}}^{(v,t)}, \mathcal{X}_{\mathcal{E}}^{(v,t)}\}$ be the set of features in frame t in video v where

  - $\mathcal{X}_{\mathcal{V}}^{(v,t)}$ is the set of node features $\{\mathbf{x}_s^{(t)} \in \mathbb{R}^{D_n} : s \in \mathcal{V}^{(v,t)}\}$ for frame t in video v.

  - $\mathcal{X}_{\mathcal{E}}^{(v,t)}$ is the set of edge features $\{\mathbf{x}_{ij}^{(t)} \in \mathbb{R}^{D_e} : (i,j) \in \mathcal{E}^{(v,t)}\}$ for frame t in video v.

- Let $\mathcal{X}_{\mathcal{T}}^{(v,t,t-1)}$ be the set of temporal features $\{\mathbf{x}_{ab}^{(t,t-1)} \in \mathbb{R}^{D_{temp}} : a \in \mathcal{V}^{(v,t)}, b \in \mathcal{V}^{(v,t-1)}\}$ between adjacent frames $t, t-1$ in video v.

- Let $\mathcal{Y}^{(v,t)} = \{\mathbf{y}_s^{(v,t)} \in \{0,1\}^L, s \in \mathcal{V}^{(v,t)} : \sum_{l=1}^{L} y_{sl}^{(v,t)} = 1\}$ be the set of labels for the nodes in frame t in video v.

D_n, D_e, D_{temp} denote the dimensions of the node, edge, and temporal features, respectively, and L denotes the number of labels. Note that compared to the notation in previous chapters, there is an additional set of temporal features *between* the superpixels in adjacent frames. In the rest of this chapter, the superscripts "v", "node", and "edge" are omitted for clarity, but the meaning should be clear from the context. The superscript t is also omitted, except when describing interactions between frames in a video.

The STRF model is shown in Figure 5.1. The top two layers correspond to a conditional restricted Boltzmann machine (CRBM) [86] with the (virtual) pooling nodes colored **orange** and the hidden nodes colored **green**. The bottom two layers correspond to a temporal SCRF. This combination of the CRBM and temporal SCRF is referred to as the STRF model. Note that if we consider the model at time t only and ignore the previous frames, we revert to the GLOC model from Chapter 4. We now describe the components of the STRF model in detail.

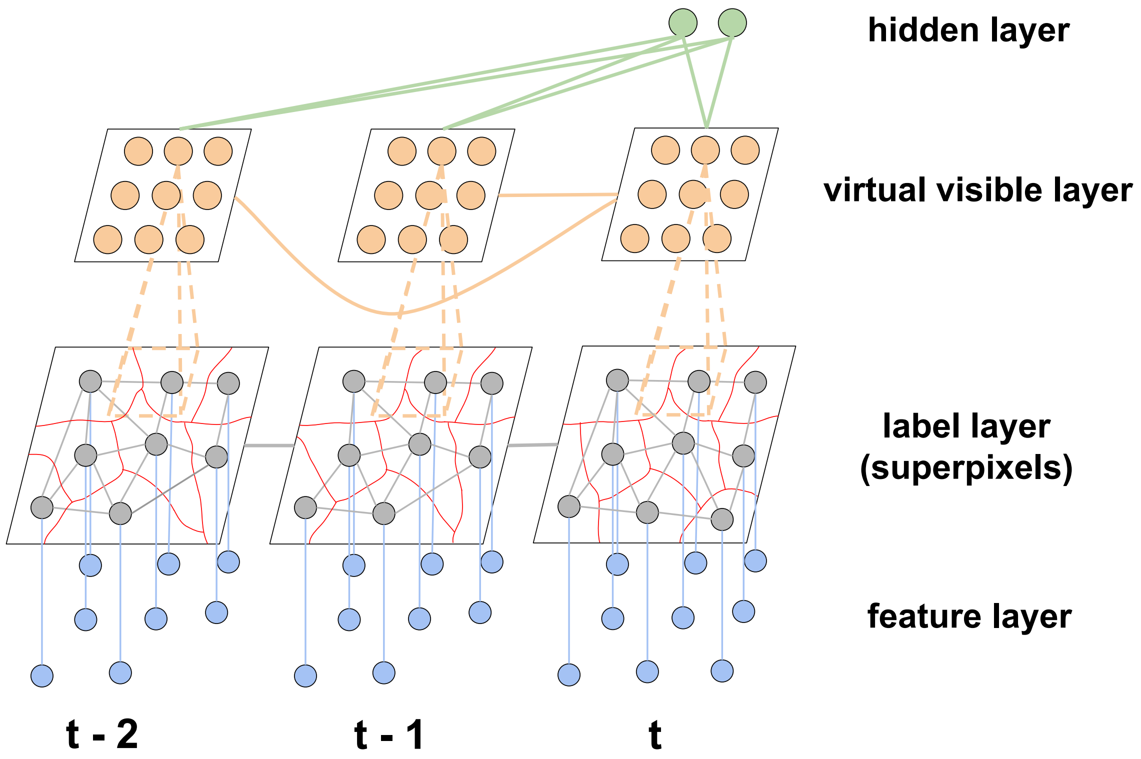


Figure 5.1. High level view of the STRF model. The model is shown for the current frame at time t and two previous frames. The top two layers correspond to the CRBM component and the bottom two layers correspond to the CRF component. The dashed lines indicate the virtual pooling between the (virtual) visible units of the CRBM and the superpixel label nodes. Parts of this model will be shown in more detail in subsequent figures. Best viewed in color.

5.3.1 RBM

The restricted Boltzmann machine (RBM) [82] is a generative model in which the nodes are arranged as a bipartite graph, consisting of a hidden layer and visible layer. The joint distribution and energy are defined as:

$$P_{\text{rbm}}(\mathcal{Y}, \mathbf{h}) \propto \exp(-E_{\text{rbm}}(\mathcal{Y}, \mathbf{h})), \tag{5.1}$$

$$E_{\text{rbm}}(\mathcal{Y}, \mathbf{h}) = -\sum_{r=1}^{R^2} \sum_{l=1}^{L} \sum_{k=1}^{K} y_{rl} W_{rlk} h_k - \sum_{k=1}^{K} b_k h_k - \sum_{r=1}^{R^2} \sum_{l=1}^{L} c_{rl} y_{rl}. \tag{5.2}$$

There are R^2 multinomial visible units, L labels, and K hidden units. $\mathbf{W} \in \mathbb{R}^{R^2 \times L \times K}$ represents the pairwise weights between the hidden units h and the visible units y, and b, c represent the biases for the hidden units and multinomial visible units, respectively. The model parameters W, b, c are trained using stochastic gradient descent. Although the exact gradient is intractable to compute, it can be approximated using contrastive divergence [37].

Virtual Pooling As before with the GLOC model in Chapter 4, a *virtual* pooling layer is used to map between the fixed grid of the RBM and the variable number of superpixels in an image. This virtual pooling is shown in Figure 5.1, as the dashed **orange** lines between the pooling and label layers. The projection matrix used for pooling is defined as

$$p_{rs} = \frac{Area(Region(s) \cap Region(r))}{Area(Region(r))},$$

where r is the index for the visible units in the RBM and s is the index for superpixels. $Region(s)$ and $Region(r)$ refer to the pixels corresponding to the superpixel s and the visible unit r. The energy function between the label nodes and the hidden nodes is now defined as

$$E_{\text{rbm}}(\mathcal{Y}, \mathbf{h}) = -\sum_{r=1}^{R^2} \sum_{l=1}^{L} \sum_{k=1}^{K} \bar{y}_{rl} W_{rlk} h_k - \sum_{k=1}^{K} b_k h_k - \sum_{r=1}^{R^2} \sum_{l=1}^{L} c_{rl} \bar{y}_{rl}, \tag{5.3}$$

where the virtual visible node $\bar{y}_{rl} = \sum_{s=1}^{S} p_{rs} y_{sl}$ are deterministically mapped from the label layer by multiplying with the projection matrix.

CRBM. While the RBM can be used to model the label shape within a particular frame of video, it may be more difficult to model temporal dependencies in the video. The conditional restricted Boltzmann machine (CRBM)[86] is an extension of the

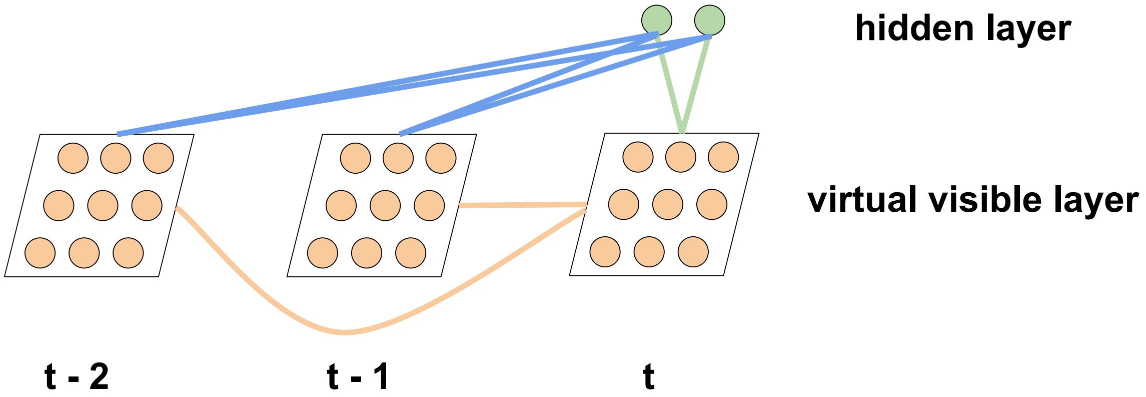
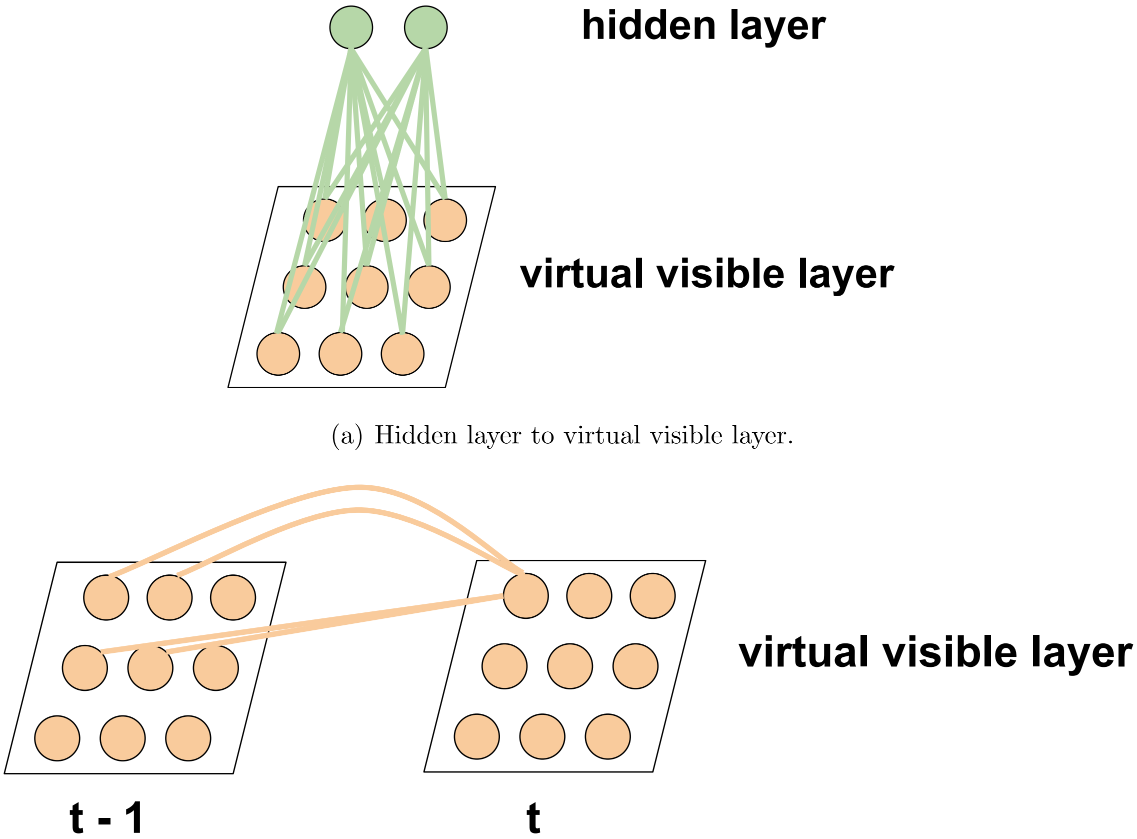


Figure 5.2. CRBM component. **Green** edges correspond to the pairwise weights W, **blue** edges correspond to the weights B, and **orange** edges correspond to the weights A in Equation (5.4). Note that in this figure, only the previous two time steps are modeled, but in the experiments, we typically model the previous three time steps. Best viewed in color.

RBM that uses previous frames in a video to act as a dynamic bias for the hidden units in the current frame. The CRBM energy is defined as

$$\begin{aligned} E_{\text{crbm}}\left(\mathcal{Y}^{(t,<t)}, \mathbf{h}^{(t)}\right) &= E_{\text{rbm}}\left(\mathcal{Y}^{(t)}, \mathbf{h}^{(t)}\right) \\ &- \sum_{w=1}^{W}\sum_{r=1}^{R^2}\sum_{l=1}^{L}\sum_{k=1}^{K} \bar{y}_{rl}^{(t-w)} B_{wrlk} h_k^{(t)} \\ &- \sum_{q=1}^{Q^2}\sum_{w=1}^{W}\sum_{r=1}^{R^2}\sum_{l=1}^{L} \bar{y}_{qrl}^{(t-w)} A_{qwrl} \bar{y}_{rl}^{(t)}, \end{aligned} \tag{5.4}$$

which includes the RBM energy $E_{\text{rbm}}\left(\mathcal{Y}^{(t)}, \mathbf{h}\right)$ defined earlier in Equation (5.3). The W frames before the current frame t act as the "history", which is always conditioned on at time t. Following the notation in [86], $\mathcal{Y}^{(<t)}$ refers to the labels of the W previous frames before the current frame. $A \in \mathbb{R}^{Q^2 \times W \times R^2 \times L}$ represents the weights of virtual visible units in the history to the current visible units and $B \in \mathbb{R}^{W \times R^2 \times L \times K}$ represents the weights of virtual visible units in the history to the hidden units. Note that there is a dense connection between the hidden units h and the virtual visible layer at each



(a) Hidden layer to virtual visible layer.

(b) Virtual visible layer at time t to $t-1$, shown for a single virtual visible node.

Figure 5.3. (a) Connections between the hidden layer to the virtual visible layer at each time step, which corresponds to an RBM. (b) Connections between the virtual visible layer at time t to $t-1$, shown for a single virtual visible node. The virtual visible node at time t in the upper-left corner is connected to a local neighborhood of size Q from the previous frame. Best viewed in color.

time step. If each time step is considered independently, this corresponds to an RBM, as shown in more detail in Figure 5.3(a).

The hidden units h are densely connected to all the virtual visible units $\bar{y}$ both in the current frame and in the history because the hidden units are meant to model global changes in object shape across time. However, the connections between virtual visible units at different time steps act as temporal smoothing and thus the interactions are likely to be more local. Thus, each visible node $\bar{y}_{rl}^{(t)}$ at time t is only connected to a local neighborhood Q in previous frames. By modeling only the local interactions between visible units the number of parameters is also significantly reduced. Figure 5.3(b) shows this local modeling for a single visible node.

The main differences between the way the CRBM is used in our model compared to its original usage in [86] are :

- The CRBM in our model is used within a discriminative framework for labeling. The goal for the CRBM is not to generate realistic data, but rather to complement the local modeling provided by the CRF and help improve labeling performance.

- The CRBM models the label shape across time, and does not model the observed features directly (which is the case in the original usage of the CRBM)

- In our model, the visible units in the current frame only model a local neighborhood (of size Q) of the visible units in the history. In contrast, the original CRBM has a dense connection between the visible units in the current frame and the visible units in the history.

5.3.2 CRF

The conditional random field [52] is a discriminative model that is used as both a baseline and a component for our later models. The CRF energy is defined as

$$E_{\text{crf}}(\mathcal{Y}, \mathcal{X}) = E_{\text{node}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}\right) + E_{\text{edge}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}\right), \tag{5.5}$$

$$E_{\text{node}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}\right) = -\sum_{s \in \mathcal{V}} \sum_{l=1}^{L} \sum_{d=1}^{D_n} y_{sl} \Gamma_{ld} x_{sd}, \tag{5.6}$$

$$E_{\text{edge}}\left(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}\right) = -\sum_{(i,j) \in \mathcal{E}} \sum_{l,l'=1}^{L} \sum_{e=1}^{D_e} y_{il} y_{jl'} \Psi_{ll'e} x_{ije}, \tag{5.7}$$

where $\Gamma \in \mathbb{R}^{L \times D_n}$ are the learned node weights and $\Psi \in \mathbb{R}^{L \times L \times D_e}$ are the learned edge weights. We use a mean-field approximation [76] (shown in Chapter 3) for approximate inference along with LBFGS optimization during learning, provided by the minFunc software [3].

5.3.2.1 Spatial CRF

As described previously in Chapter 3, the spatial CRF (SCRF) is a variant of the CRF in which a different set of weights is learned for each position in an image. Empirically this was found to perform better and so the SCRF is used instead of the CRF as both a baseline and a component for later models. The energy function for nodes is revised to:

$$E_{\text{node}'}(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}) = -\sum_{s \in \mathcal{V}} \sum_{l=1}^{L} y_{sl} \sum_{n=1}^{N^2} p_{sn} \sum_{d=1}^{D_n} \Gamma_{ndl} x_{sd} \tag{5.8}$$

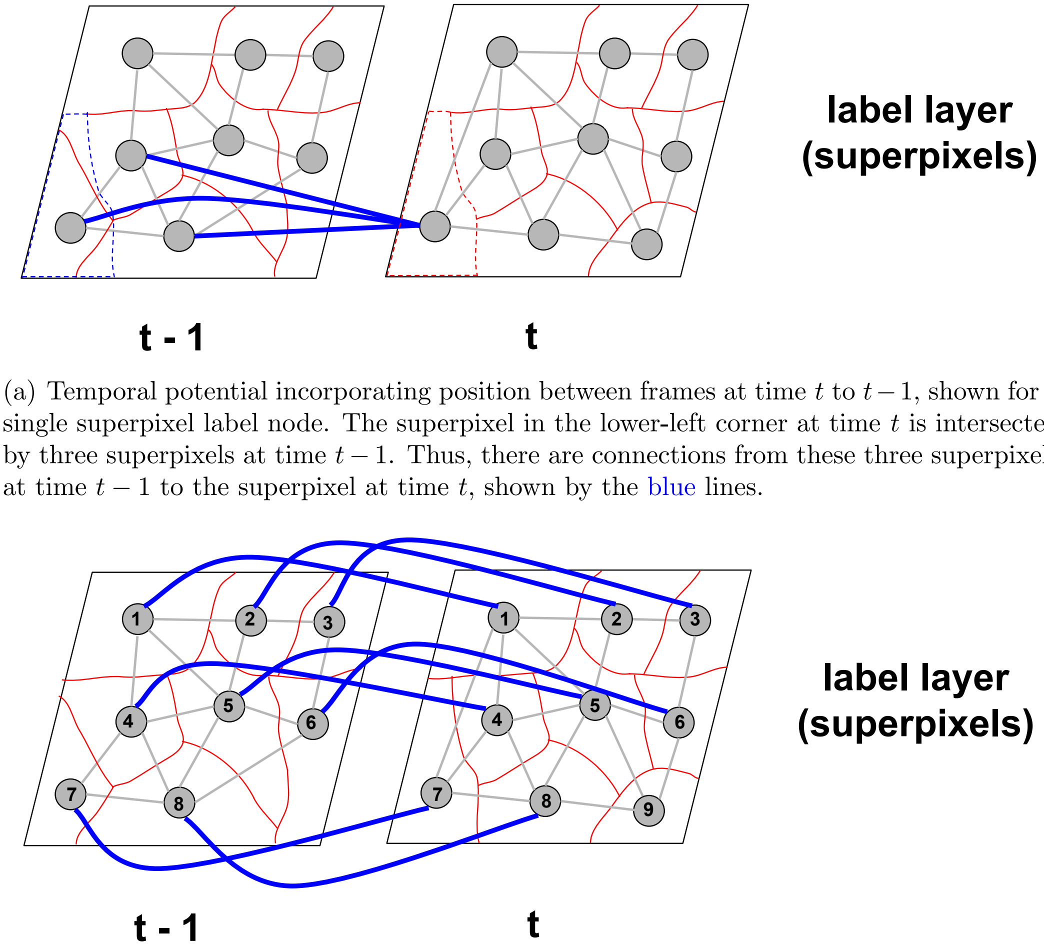
where $\Gamma \in \mathbb{R}^{N^2 \times D \times L}$ are the learned node weights and the total energy is revised to

$$E_{\text{scrf}}(\mathcal{Y}, \mathcal{X}) = E_{\text{node'}}(\mathcal{Y}, \mathcal{X}_{\mathcal{V}}) + E_{\text{edge}}(\mathcal{Y}, \mathcal{X}_{\mathcal{E}}) \,. \tag{5.9}$$

In particular, the image is divided into an $N \times N$ grid and a set of node weights are learned specific to each cell in the grid. A projection matrix $\{p\}$ is used to map between the superpixels and the grid, in a similar way to the virtual pooling in the RBM. Inference for the SCRF is very similar to inference for the CRF as described in Chapter 3.

5.3.2.2 Temporal SCRF

One way to extend a traditional CRF model for labeling in videos is to incorporate temporal potentials, which has been applied to tasks such as labeling [92, 27, 4] and activity recognition [81, 91]. In some models, during inference at time t, the temporal potentials only look at previous frames while other models allow for interactions with future frames. In addition, models that incorporate temporal potentials typically look



(a) Temporal potential incorporating position between frames at time t to $t-1$, shown for a single superpixel label node. The superpixel in the lower-left corner at time t is intersected by three superpixels at time $t-1$. Thus, there are connections from these three superpixels at time $t-1$ to the superpixel at time t, shown by the blue lines.

(b) Temporal potential incorporating TSP ID between frames at time t to $t-1$. Superpixels 1-8 exist at both time t and $t-1$, and thus there is a connection (indicated by blue lines) between a superpixel at time $t-1$ to its corresponding superpixel at time t. However, superpixel 9 is “created” at time t and thus there is no connection from the previous frame.

Figure 5.4. Temporal potentials. Best viewed in color.

in a window around the current frame at time t, rather than the entire sequence, which helps to keep the inference tractable. In our model, temporal potentials look only at the previous frame and are used to encourage smoothing between adjacent frames in a video in much the same way that edge potentials encourage spatial smoothing within an image. Two types of temporal potentials are used:

- **Position smoothness**: This potential encourages a consistent labeling between superpixels in adjacent frames that are approximately in the same position and have similar color and texture. The energy is defined as

$$E_{\text{tpot1}}\left(\mathcal{Y}^{(t,t-1)}, \mathcal{X}_{\mathcal{T}}^{(t,t-1)}\right) = -\sum_{a\in\mathcal{V}^{(t)}} \sum_{b\in Int(V^{(t-1)},a)} \sum_{l,l'=1}^{L} \sum_{e=1}^{D_{temp}} y_{al}^{(t)} y_{bl'}^{(t-1)} \Phi_{ll'e} x_{abe}^{(t,t-1)}, \tag{5.10}$$

where $\Phi \in \mathbb{R}^{L\times L\times D_{temp}}$ represent the temporal weights, and $Int(\mathcal{V}^{(t-1)}, a)$ refers to superpixels in frame $t-1$ that intersect with superpixel a in the current frame t. Thus, only superpixels that intersect with superpixel a in the previous frame are counted in this potential. Figure 5.4(a) shows the connections for this temporal potential for a single superpixel node. The figure shows the superpixel in the lower-left corner at time t and its projection at time $t-1$ (shown in dotted **blue** lines). At time $t-1$, there are three superpixels that are intersected by the dotted blue lines. Thus, there are connections from these three superpixels at time $t-1$ to the superpixel at time t, shown by the solid **blue** lines.

- **Superpixel smoothness**: Temporal superpixels (TSP) [12] are used to segment the frames in a video. They have the desirable property of maintaining their position on an object through time. For example, a TSP on a person's cheek will stay "stuck" to the person's cheek as long as the person's pose does not change significantly (e.g. the person does not move their head). For our task, these TSPs have been found empirically to be very pure in the sense that a TSP tends to remain a single label for most of its lifetime. The following temporal potential is used to encourage consistent labeling between the same TSPs in adjacent frames,

$$E_{\text{tpot2}}\left(\mathcal{Y}^{(t,t-1)}\right) = -\sum_{a\in\mathcal{V}^{(t)}} \sum_{b\in\mathcal{V}^{(t-1)}} \sum_{l,l'=1}^{L} y_{al}^{(t)} y_{bl'}^{(t-1)} \Pi_{ll'} \left[a=b\right], \tag{5.11}$$

where $\Pi \in \mathbb{R}^{L\times L}$ represent the temporal weights and $[a = b]$ denotes indicator notation checking whether superpixel a is equal (i.e. has the same TSP ID) to superpixel b. Figure 5.4(b) shows the connections of this temporal potential at time t and $t-1$. Note that superpixels 1-8 exist at both time t and $t-1$, and thus there is a connection (indicated by **blue** lines) between a superpixel at time $t-1$ to its corresponding superpixel at time t. However, superpixel 9 is "created" at time t and therefore there is no connection from the previous frame.

Incorporating these temporal potentials, the energy for the temporal SCRF model is defined as

$$E_{\text{tscrf}}(\mathcal{Y}^{(t,t-1)}, \mathcal{X}^{(t,t-1)}) = E_{\text{scrf}}\left(\mathcal{Y}^{(t)}, \mathcal{X}^{(t)}\right) + E_{\text{tpot1}}\left(\mathcal{Y}^{(t,t-1)}, \mathcal{X}_{\mathcal{T}}^{(t,t-1)}\right) + E_{\text{tpot2}}\left(\mathcal{Y}^{(t,t-1)}\right), \tag{5.12}$$

where the SCRF energy defined earlier is simply augmented by the temporal potentials.

Inference. Inference using the temporal SCRF is described in Algorithm 5. For the first frame (time $t = 1$), the SCRF is used for inference, since it does not depend on previous frames. Afterward, inference in the temporal SCRF is computed using a mean-field approximation in which the temporal potentials and label guesses from the previous frame (denoted as $\boldsymbol{\alpha}$) are used for inference at time t. In step 1, the variational parameters $\boldsymbol{\mu}^{(0)}$ are initialized to the logistic regression guess, which depends only on node potentials. In step 2, the temporal potentials from the previous frame $t-1$ are computed using label guesses from the previous frame. Recall that $Int(\mathcal{V}^{(t-1)}, s)$ refers to superpixels in the previous frame that intersect with superpixel s in the current frame. In step 4, the node, edge, and temporal potentials are used together to update the parameters $\boldsymbol{\mu}^{(i)}$. The algorithm then iterates until the param-

Algorithm 5 Mean-Field inference for the temporal SCRF

```
1: Initialize $\boldsymbol{\mu}^{(0)}$ as follows:
       $$\mu_{sl}^{(0)} = \frac{\exp\left(f_{sl}^{\text{node}}\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}}\right)}$$
   where
       $$f_{sl}^{\text{node}}\left(\mathcal{X}_{\mathcal{V}}, \{p_{sn}\}, \Gamma\right) = \sum_{n,d} p_{sn} x_{sd} \Gamma_{ndl}$$
2: Let $\boldsymbol{\alpha}$ be the mean-field estimates from the previous frame where
       $$f_{sl}^{\text{temp}}\left(\boldsymbol{\alpha}; \mathcal{X}_{\mathcal{T}}^{(t,t-1)}, \Phi, \Pi\right) = \sum_{b \in Int(V^{(t-1)},s),} \sum_{l,l'=1}^{L} \sum_{e=1}^{D_{temp}} \alpha_{bl'} \Phi_{ll'e} x_{sbe}$$
       $$+ \sum_{b \in \mathcal{V}^{(t-1)}} \sum_{l,l'=1}^{L} \alpha_{bl'} \Pi_{ll'} \left[s = b\right]$$
3: for i=0:$MaxIter$ (or until convergence) do
4:     update $\boldsymbol{\mu}^{(i+1)}$ as follows: $\mu_{sl}^{(i+1)} =$
           $$\frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl}^{\text{temp}}\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl'}^{\text{temp}}\right)}$$
       where
           $$f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}; \mathcal{X}_{\mathcal{E}}, \mathcal{E}, \Psi\right) = \sum_{j:(s,j) \in \mathcal{E}} \sum_{l',e} \mu_{jl'} \Psi_{ll'e} x_{sje}$$
5: end for
```

eters $\boldsymbol{\mu}^{(i)}$ either no longer change or a maximum number of iterations ($MaxIter$) is reached.

There is not much additional cost for inference (compared to inference in the SCRF) because the labels from the previous frame $t-1$ are assumed fixed, and thus the temporal potentials only need to be computed once. In step 4, the node and temporal potentials are included in the update for $\boldsymbol{\mu}^{(i)}$ but only the edge potentials change during iteration. Average inference time per frame for the temporal SCRF is about 0.78 (sec) compared to about 0.74 (sec) for the SCRF, on an Intel i7.

Learning. The parameters in the model are $\{\Gamma, \Psi, \Phi, \Pi\}$. Recall that Γ, Ψ are the weights for the node and edge weights respectively, and Φ, Π are the weights for the temporal potentials. The parameters are trained to maximize the conditional log-likelihood of the training data $\{\mathcal{Y}^{(m)}, \mathcal{X}^{(m)}\}_{m=1}^{M}$,

$$L = \max_{\Gamma,\Psi,\Phi,\Pi} \sum_{m=1}^{M} \log P_{\text{tscrf}}(\mathcal{Y}^{(m)}|\mathcal{X}^{(m)}),$$

where M is the number of labeled videos in the training set. The model parameters are learned using the LBFGS optimization in conjunction with the minFunc [3] software package.

5.3.3 Shape-Time Random Field

The GLOC model (presented in Chapter 4) incorporates a strong, global shape prior for the semantic labeling of images. The Shape-Time Random Field (STRF) is an extension of the GLOC model for the semantic labeling of videos which incorporates both temporal smoothing (using the temporal SCRF) and temporal shape dependencies (using the CRBM). The conditional distribution and energy of the STRF model are defined as:

$$P_{\text{strf}}(\mathcal{Y}^{(t)}|\mathcal{Y}^{(<t)}, \mathcal{X}^{(t,t-1)}) \propto \sum_{\mathbf{h^{(t)}}} \exp\left(-E_{\text{strf}}(\mathcal{Y}^{(t,<t)}, \mathcal{X}^{(t,t-1)}, \mathbf{h^{(t)}})\right), \tag{5.13}$$

$$E_{\text{strf}}\left(\mathcal{Y}^{(t,<t)}, \mathcal{X}^{(t,t-1)}, \mathbf{h}^{(t)}\right) = E_{\text{tscrf}}\left(\mathcal{Y}^{(t,t-1)}, \mathcal{X}^{(t,t-1)}\right) + E_{\text{crbm}}\left(\mathcal{Y}^{(t,<t)}, \mathbf{h}^{(t)}\right), \tag{5.14}$$

The complete model is shown in Figure 5.1. The goal is to use the CRBM (top two layers) to provide a dynamic bias for the hidden units, based on previous history, to help with the temporal SCRF label classification (bottom two layers).

5.3.3.1 Inference

There is a spectrum of approaches available when performing inference for the STRF model, depending on the amount of approximation used. One end of the spectrum corresponds to using a large degree of approximation. In the extreme case,

Algorithm 6 Mean-Field approximate inference for the STRF model at time t

```
Initialize $\boldsymbol{\mu}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$ as follows:
    $\mu_{sl}^{(0)} = \frac{\exp\left(f_{sl}^{\text{node}}\right)}{\sum_{l'}\exp\left(f_{sl'}^{\text{node}}\right)}$
    $\gamma_k^{(0)} = sigmoid\left(\sum_{r,l}\left(\sum_{s} p_{rs}\mu_{sl}^{(0)}\right)W_{rlk} + b_k\right)$
  where
    $f_{sl}^{\text{node}}\left(\mathcal{X}_{\mathcal{V}}, \{p_{sn}\}, \Gamma\right) = \sum_{n,d} p_{sn}x_{sd}\Gamma_{ndl}$
Let $\boldsymbol{\alpha}$ be the mean-field estimates from the previous frame where
    $f_{sl}^{\text{temp}}\left(\boldsymbol{\alpha}; \mathcal{X}_{\mathcal{T}}^{(t,t-1)}, \Phi, \Pi\right) = \sum_{b\in Int(V^{(t-1)},s),} \sum_{l,l'=1}^{L} \sum_{e=1}^{D_{temp}} \alpha_{bl'}\Phi_{ll'e}x_{sbe} + \sum_{b\in\mathcal{V}^{(t-1)}} \sum_{l,l'=1}^{L} \alpha_{bl'}\Pi_{ll'}\left[s=b\right]$
    $f_{sl}^{\text{crbm}}\left(\boldsymbol{\gamma}; \{p_{rs}\}, \mathbf{W}, \mathbf{C}, \mathbf{A}, \boldsymbol{\alpha}\right) = \sum_{r} p_{rs}\left(c_{rl} + \sum_{k} W_{rlk}\gamma_k + \sum_{q,w} A_{qwrl}\left(\sum_{s'} p_{rs'}^{(t-w)}\alpha_{s'l}^{(t-w)}\right)\right)$
for i=0:$MaxIter$ (or until convergence) do
    update $\boldsymbol{\mu}^{(i+1)}$ as follows: $\mu_{sl}^{(i+1)} =$
        $\frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl}^{\text{temp}} + f_{sl}^{\text{crbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}{\sum_{l'}\exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl'}^{\text{temp}} + f_{sl'}^{\text{crbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}$
      where
        $f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}; \mathcal{X}_{\mathcal{E}}, \mathcal{E}, \Psi\right) = \sum_{j:(s,j)\in\mathcal{E}} \sum_{l',e} \mu_{jl'}\Psi_{ll'e}x_{sje}$
    update $\boldsymbol{\gamma}^{(i+1)}$ as follows:
        $\gamma_k^{(i+1)} = sigmoid\left(b_k + \sum_{r,l}\left(\sum_{s} p_{rs}\mu_{sl}^{(i+1)}\right)W_{rlk} + \sum_{w,r,l}\left(\sum_{s'} p_{rs'}^{(t-w)}\alpha_{s'l}^{(t-w)}\right)B_{wrlk}\right)$
end for
```

each video frame can be considered independently of the other frames. Further along the spectrum, using less approximation, inference of a frame at time t can depend on the previous frame at time $t-1$ (as in the case of the temporal SCRF) or perhaps a window of previous frames. The other end of the spectrum corresponds to using a low degree of approximation. In the extreme case, we can perform inference using *all* frames together, which corresponds to a large graph connecting all frames. This may involve performing multiple forward and backward passes through the sequence of video frames, until convergence.

Practically, it is preferable to avoid both ends of the spectrum. In the first case, by treating each frame independently we ignore potentially useful information either earlier or later in the sequence, which may lead to undesirable labeling discontinuities. In the second case, using less approximation and treating the video sequence as one large graph may be the "correct" way to perform inference, but it is likely to be computationally prohibitive due to the many forward and backward passes required.

Our goal is to find a point on this spectrum to balance using less approximation while also being computationally efficient. We decided to employ a feed-forward inference procedure which depends only on a window of W previous frames at time t. This approach is computationally efficient since the history of W previous frames is *fixed* at time t, and so the only latent variables at time t are the hidden units of the CRBM and the label variables. It is possible that this feed-forward inference may ignore important information later in the sequence that may be useful, but the extra backward propagation steps may be computationally prohibitive. In addition, this feed-forward approach may be appropriate in a real-time setting such as surveillance.

In particular, the parameter W determines how many previous frames are used to serve as the history when performing inference for the current frame at time t. During inference, the first W frames are computed using the GLOC [47] model, which does not depend on previous frames. Afterward, inference proceeds in a sliding window fashion as described in Algorithm 6. We again use a mean-field approximate inference approach. In step 1, the variational parameters $\boldsymbol{\mu}^{(0)}$ are initialized to the logistic regression guess (which depends only on node potentials) and $\boldsymbol{\gamma}^{(0)}$ are initialized using $\boldsymbol{\mu}^{(0)}$. Step 2 computes the temporal and CRBM potentials from previous frames in the history. The algorithm then iterates between updating the parameters $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\gamma}^{(i)}$, until either a maximum number of iterations is reached ($MaxIter$) or the parameters no longer change after updates. Note that steps 4 and 5 use the mean field estimates $\boldsymbol{\alpha}^{(t-w)}$ from previous frames in the history, where $p^{(t-w)}$ denotes the corresponding projection matrix from previous frames.

In addition, we also use another parameter S to determine how many frames to skip in the history. For example, if $W = 3$ and $S = 2$, then out of the previous 6 frames, every other frame is used in the history. Skipping some frames may still allow us to model the temporal dependencies properly since there may not be a large change between consecutive frames $t-1$ and $t-2$. In addition, skipping frames in the history allows us to use a larger window while still keeping the number of parameters tractable.

5.3.3.2 Learning

The STRF model is learned using a piecewise learning scheme. That is, the temporal SCRF and CRBM components are learned separately and then a scalar parameter λ is used to weight the contribution between them. In our experiments, we tried a variety of λ values between $\{0..1\}$ and chose λ based on which value performed best on the validation set. The piecewise model replaces the the node update $\mu_{sl}^{(i+1)}$ in step 4 in Algorithm 6 with

$$\mu_{sl}^{(i+1)} = \frac{\exp\left(f_{sl}^{\text{node}} + f_{sl}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl}^{\text{temp}} + \lambda f_{sl}^{\text{crbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}{\sum_{l'} \exp\left(f_{sl'}^{\text{node}} + f_{sl'}^{\text{edge}}\left(\boldsymbol{\mu}^{(i)}\right) + f_{sl'}^{\text{temp}} + \lambda f_{sl'}^{\text{crbm}}\left(\boldsymbol{\gamma}^{(i)}\right)\right)}.$$

It is possible that jointly training all the model parameters $\{\Gamma, \Psi, \Phi, \Pi, A, B\}$ may perform better than a piecewise model. However, as shown from experiments using the GLOC model in Chapter 4, the piecewise model can still offer good performance, even though it may not match the performance of the jointly trained version.

5.4 Data

Our models are evaluated on videos from the YouTube Faces Database [97] (YFDB), which is a large database of "real world" videos found on YouTube, and not taken from a controlled, laboratory setting. Videos from YFDB contain a large variety of face shapes, poses, lighting conditions and occlusions, making them challenging to label. Table 5.1 shows frames from a video from YFDB, including the corresponding superpixel and ground truth labeling for each frame.

We randomly selected 100 videos from YFDB and among these videos, 20 consecutive frames were manually labeled for Hair/Skin/Background regions. There are some people that have multiple videos in YFDB and so we required that the 100 videos were of unique people. This is to ensure that when using cross-validation for experiments, the same person is not used for both training and testing.

5.4.1 Alignment

Previously, for the experiments with the GLOC model in Chapter 4, images were first aligned using a congealing algorithm [43] before being segmented into superpixels. For videos, we tried several alignment algorithms (including one provided by YFDB) and found that in many cases, they result in an unstable, coarse alignment.

Some cases showing the significant scale differences between frames and other transformation instabilities, are shown in Tables 5.2 and 5.3. These figures show the YFDB-provided alignment and two other alignment approaches: (1) a pre-learned deep funnel using the method of [44], and (2) a pre-learned SIFT-congealed funnel using the approach of [43]. Both funnels were pre-learned on LFW images and the the SIFT-congealed funnel was used to attain the image alignments for the previous GLOC experiments. By aligning each frame in the video to a canonical position (using a funnel), the video itself should also be aligned.

Model	t	t+2	t+4	t+6	t+8	t+10
Original						
Mean VJ						
YFDB Alignment						
Deep Alignment						
SIFT Congealing						

Table 5.2. Comparison of alignment algorithms. This figure shows the result of several alignment algorithms for a video clip from YFDB. Every other frame is shown for a sequence of 10 frames. The original VJ face detection (row 1) has some temporal instability but using the simple approach of fixing the height and width of the detected face box in each frame to the mean height and width of the face boxes for the video (row 2) results in a fairly stable and smooth sequence.

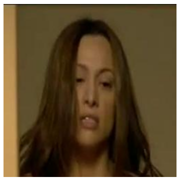
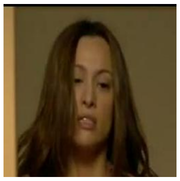
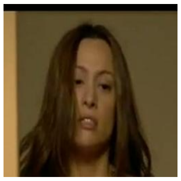
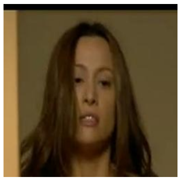
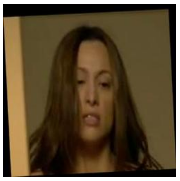
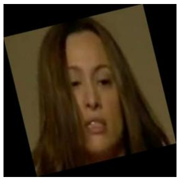
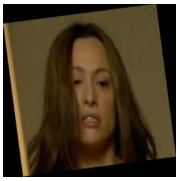
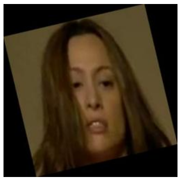
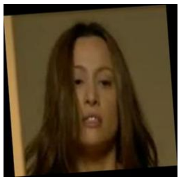
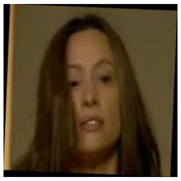
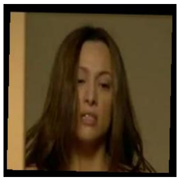
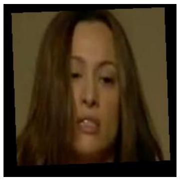
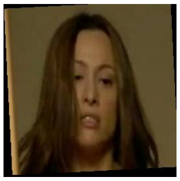
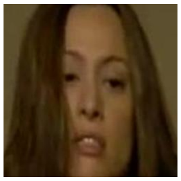
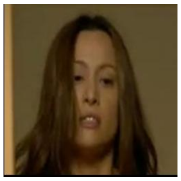
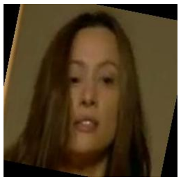
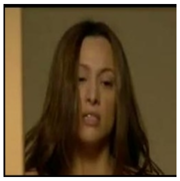
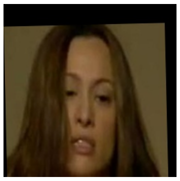
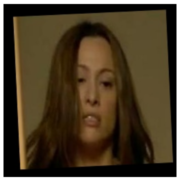
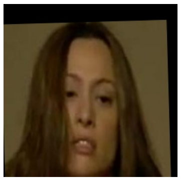
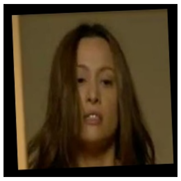
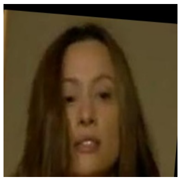
Model	t	t+2	t+4	t+6	t+8	t+10
Original						
Mean VJ						
YFDB Alignment						
Deep Alignment						
SIFT Congealing						

Table 5.3. Comparison of alignment algorithms. This figure shows the result of several alignment algorithms for a video clip from YFDB. Every other frame is shown for a sequence of 10 frames. Note that the alignment algorithms (in rows 3-5) can produce significant transformation instabilities such as scale differences.

Model	t	t+2	t+4	t+6	t+8	t+10
Original						
Mean VJ						
YFDB Alignment						
Deep Alignment						
SIFT Congealing						

Table 5.4. Comparison of alignment algorithms. This figure shows the result of several alignment algorithms for a video clip from YFDB. Every other frame is shown for a sequence of 10 frames. In this case, the alignment approaches work well and result in a stable, temporally smooth sequence. The YFDB alignment results in images slightly smaller in scale compared to alignments by other approaches.

There are some transformations, especially by the YFDB-provided alignment, that result in excessive rotation such as in frame $t+10$ in Table 5.2. In addition, there are significant scale differences between frames such as in frames $t+4, t+6, t+10$ using the deep alignment in Table 5.3. It may be possible to smooth the transformations provided by these alignment methods using post-processing provided by a model such as a Kalman filter [48]. However, judging by the large scale differences present and other instabilities, we felt that much of the instability would still remain even after post-processing.

In some cases, the alignment approaches work well, as shown for example, in Table 5.4. In this case, the YFDB alignment results in images that are slightly smaller in scale compared to other results, but are still temporally smooth. Both the deep alignment and SIFT congealing approaches work well and result in stable, temporally smooth sequences.

Overall, the deep alignment and SIFT congealing algorithms do not appear to work as well on YFDB videos as for LFW images, which was their original application. It is possible there is some fundamental difference between LFW images and YFDB video frames which prevents applying a funnel learned from LFW images directly to YFDB videos. It is also possible there may be slight differences in the alignment implementations we used and the original implementations.

We resorted to a simpler approach to avoid using an unstable alignment. Because it is preferable to train the CRBM over temporally smooth data, we used the output of the Viola Jones face detector [88], but fixed the height and width of the detected face box to the mean height and width of the detected face boxes for all frames in the video. Then, for each frame in the video, a bounding box for the face is cropped out using the center of the Viola Jones detection (provided by YFDB) using the dimensions of the mean width and height for the video. Following the process of LFW [46], the bounding box is expanded by a factor of 2.2 in each direction and then resized to 250 × 250 pixels. This simple fix tends to produce a stable, temporally smooth set of frames, as seen in Tables 5.2 and 5.3. In these tables, the original VJ face detection (row 1) has some temporal instability but using the simple approach of fixing the width and height for the video (row 2) results in a fairly stable and smooth sequence.

5.4.2 Superpixel Segmentation

After processing the YFDB videos, the video frames are segmented using temporal superpixels (TSP) [12]. Using TSPs for our task is more appropriate than frame-independent superpixels because TSPs can maintain temporal consistency across frames in a video. That is, a TSP on a person's cheek will stay "stuck" to the person's cheek as long as the person's pose doesn't change significantly (e.g. the person does not move their head). In addition, for YFDB data, TSPs tend to be very pure in that TSPs will rarely change labels. If a TSP is initially labeled as hair, it will tend to stay labeled as hair throughout its lifetime. The TSP code generated about 300-400 superpixels per frame. There are alternatives to TSP such as supervoxel approaches [32, 100, 99], but TSPs were typically found to have a longer lifetime than supervoxels, are more uniform in size, and maintain better label consistency across time than supervoxels.

5.5 Experiments

As mentioned previously in Section 5.4, 100 videos were randomly selected from YFDB and for each video a "chunk" of 20 consecutive frames was manually labeled for Hair/Skin/Background regions. This resulted in a labeled database with a total of 2000 labeled frames. For the experiments, the labeled data is divided into 5 equal, disjoint sets for use in cross-validation. For each of the 5 cross-validation sets, 3 of the folds are used for training, 1 for validation and 1 for testing. There is only one instance of each person in the 100 videos, and so the same person is never used in both training and testing.

5.5.1 Features

The following features are generated for each superpixel and are the same features used in Chapter 4 and in [45, 47].

- **Color**: Normalized histogram over 64 bins generated by running K-means over pixels in LAB space. Image pixels are first clustered using K-means (using K = 64), and each pixel is assigned to its closest centroid. Afterward a normalized histogram is computed using all the pixel assignments within a superpixel.

- **Texture**: Normalized histogram over 64 textons which are generated according to [63]. To generate textons, we first convolve a set of training images with a filterbank and gather the filter responses. In our experiments, the filterbank consists of 12 filters at varying orientations, and at 3 different scales for a total of 36 filters. Specifically, the 3 sets of filters were of dimensions $19 \times 19, 27 \times 27$, and 39×39. Each pixel in the image now has a corresponding vector of 36 filter responses. These filter responses are then clustered using K-means into bins or *textons* (in our case we cluster into 64 textons). For a new image, the image is convolved with the filterbank to get filter responses for each pixel, and these pixel responses are then assigned to the closest texton. Afterward, a normalized histogram is computed for all the pixel assignments within a superpixel.

Note that position features are not included here, as they were in Chapter 4. In previous experiments with the GLOC model, the spatial CRF (which does not use position features) outperformed the CRF and so we decided to use the SCRF as the baseline instead of the CRF. The following set of edge features are computed between a pair of adjacent superpixels, within an image.

- **Probability of Boundary (Pb)** [64]: Sum of the Pb values between adjacent superpixels.
- **Color**: L2 distance between color histograms for adjacent superpixels.
- **Texture**: Chi-squared distance between texture histograms for adjacent superpixels as computed in [45, 47]

$$\chi^2 = \frac{1}{2}\sum_{i=1}^{64}\frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)},$$

  where h_1, h_2 refer to the texture histograms of adjacent superpixels.

These edge features are also used for the temporal potentials in between adjacent frames, except for the Pb feature. It is unclear how to incorporate the Pb feature, which is defined spatially within a frame, in this temporal manner.

5.5.2 Evaluation

Table 5.5 shows the results of the progression of models from the baseline SCRF to our STRF model. The results shown are the test set results for all five cross-validation folds together.

- **SCRF**. As mentioned in Chapter 4, the SCRF was empirically observed to have significantly better performance than the CRF and so the SCRF is used as a baseline instead of the CRF. In addition to the labeled training images, all 1500 labeled LFW training images from the Part Labels Database[1] are included as well. This resulted in a total training set size of $60 \times 20 + 1500 = 2700$ images, since each cross-validation training set consists of 60 videos with 20 frames in each video.

[1] vis-www.cs.umass.edu/lfw/part_labels/

- **SCRF + Temporal**. Temporal potentials are added to the SCRF model and trained jointly as described in Section 5.3.2.2.

- **SCRF + RBM**. This is the GLOC model [47] presented in Chapter 4. However, the GLOC model used here is a piecewise model in which the RBM and SCRF components are trained separately and then combined together using a scalar tradeoff parameter λ found using the validation set.

  We also trained a joint GLOC model using available code [1] (again adding all 1500 LFW training images to each fold) but this resulted in lower performance compared to the piecewise model. We used the default parameters to train the GLOC model, suggesting that the jointly trained GLOC model may be sensitive to its choice of hyperparameters, which may have contributed to this drop in performance.

- **SCRF + RBM + Temporal**. This model consists of the jointly trained SCRF + Temporal model defined earlier, but with the added contribution from the RBM, combined in a piecewise way.

- **SCRF + CRBM**. The CRBM is added to the SCRF and combined in a piecewise model.

- **STRF**. The complete model combines the jointly trained SCRF + Temporal model and the CRBM in a piecewise model.

The number of hidden units is set to $K = 400$. A grid size of $N = 16$ is used for the spatial CRF and a grid size of $R = 32$ is used for the RBM. For the piecewise models, typical values of λ were between 0.5 to 0.7 (varying slightly for each fold). For the CRBM component, a window size of $W = 3$ is used and $S = 1$ and $S = 3$ are used as values for the number of previous frames to skip over. The size of the local neighborhood is set to $Q = 3$, which corresponds to the local modeling done between virtual visible layers at time t and previous time steps. For each fold, the hyperparameters were chosen based on what setting performed best on the validation set. Regarding computation times, on a multicore Intel i7, the average inference time for the STRF model is about 0.357 seconds per frame, which is slightly more than the GLOC model which takes about 0.334 seconds per image.

Table 5.5 shows the results of cross-validation for all models. The following metrics are used (with respect to superpixels):

- **Error reduction**: computed as the following, with respect to the SCRF baseline:

$$\text{error reduction(model)} = \frac{[1 - \text{accuracy(SCRF)}] - [1 - \text{accuracy(model)}]}{1 - \text{accuracy(SCRF)}}.$$

- **Overall accuracy**: the number of superpixels classified correctly divided by the total number of superpixels.

- **Category-specific superpixel accuracy**: for each class, the number of superpixels classified correctly divided by the total number of superpixels.

- **Category average**: average of the category-specific accuracies.

The mean and standard error of the mean (SEM) for these metrics are reported for all models. In addition, we computed two-sided paired t-tests for STRF compared with all other models.

By evaluating the results of different models, we can observe the effects of adding different components such as temporal potentials and the CRBM to the baseline SCRF. For example, adding temporal potentials to the baseline SCRF seems to help, almost as much as adding the shape prior, as shown by improvements in the mean error reduction and mean overall accuracy. We can also compare the effects of adding the RBM and CRBM components. The SCRF+RBM (GLOC [47]) and

	Error	Overall				Category
Model	Reduction	Accuracy	Hair	Skin	BG	Average
S	0.000 ± 0.000	0.903 ± 0.010	***0.649 ± 0.030***	0.892 ± 0.014	0.952 ± 0.006	***0.831 ± 0.014***
S+T	0.032 ± 0.022	0.906 ± 0.010	**0.681 ± 0.030**	***0.888 ± 0.015***	0.952 ± 0.007	***0.840 ± 0.013***
S+R [47]	***0.044 ± 0.016***	***0.907 ± 0.011***	0.613 ± 0.038	***0.907 ± 0.012***	***0.960 ± 0.006***	0.827 ± 0.015
S+R+T	***0.089 ± 0.026***	***0.911 ± 0.011***	0.644 ± 0.038	***0.907 ± 0.014***	**0.961 ± 0.006**	***0.837 ± 0.015***
S+C	***0.059 ± 0.017***	***0.909 ± 0.008***	***0.660 ± 0.026***	***0.904 ± 0.012***	0.955 ± 0.006	***0.840 ± 0.011***
S+C+T	**0.110 ± 0.027**	**0.914 ± 0.009**	***0.678 ± 0.038***	**0.911 ± 0.011**	***0.956 ± 0.007***	**0.848 ± 0.014**

Table 5.5. Labeling performance. All metrics are with respect to superpixels. Model components are defined as (S): SCRF, (T): Temporal, (R): RBM, (C): CRBM. For each model, the mean and standard error of the mean (SEM) are given for each metric (from cross-validation). For each metric, the result in **blue** indicates the best performing model and results in ***italics*** indicate models with performances not statistically significantly different from the best model at the $p = 0.05$ level as measured by a two-sided paired t-test. Numbers in regular typeface indicate results that are significantly different from the best model.

SCRF+CRBM models have similar mean overall accuracies but SCRF+CRBM outperforms SCRF+RBM for the mean hair accuracy by about 4.7%, leading to a better mean category average. SCRF+CRBM also outperforms SCRF+RBM for mean error reduction.

The STRF model is composed of both temporal potentials and the CRBM component, along with the baseline SCRF. STRF results in a mean error reduction over the baseline SCRF by about 11%. In terms of mean scores, STRF outperforms other models for the following metrics: error reduction, overall accuracy, skin class, and category average (but we cannot claim these differences are statistically significant). STRF does have a significant improvement in mean error reduction and mean overall accuracy over the SCRF+Temporal model. In addition STRF has a significant improvement over SCRF+RBM for mean category average. We note that the baseline SCRF model had already achieved about 90% accuracy, and it may be increasingly difficult to make large gains. In addition, many of the changes made by the STRF model are subtle improvements which may not result in large gains in accuracy.

Tables 5.6- 5.8 show successful examples for the STRF model and Tables 5.9, 5.10 show typical failure cases. There is a noticeable improvement by the STRF model in Table 5.6 in labeling the right side of the woman's hair, which is missed by other models. In this case, the SCRF+CRBM model captured some of the correct hair shape at times t and $t+2$, but then incorrectly labeled the hair shape in later frames. It is possible that the temporal potentials were important for this example, to help "carry over" the hair shape from previous frames. In addition, Table 5.7 shows a more subtle improvement made by the STRF model that captures the hair on both sides of the woman's face. The SCRF+Temporal also manages to label the hair region but also generates an irregular hair shape, possibly due to the lack of a shape prior. Table 5.8 shows another subtle improvement by the STRF model which captures the skin region around the woman's neck, which is consistently missed by other models. Typical failure cases are shown in Tables 5.9, 5.10 in which the models with temporal potentials (including the STRF model) consistently generate incorrect hair shapes possibly because previous errors in hair shape are propagated through time.

Overall, adding the CRBM and temporal potentials result in both qualitative and quantitative improvements over baseline models. Adding either component separately also results in improvements but adding both together resulted in larger improvements. However, in some cases, the temporal potentials can incorrectly propagate errors in label shape (such as the hair shape). It is possible this error propagation may be mitigated by incorporating information from future frames. Earlier, we discussed a spectrum of approaches for performing inference in the STRF model. In particular, we adopted a feed-forward approach in which a window of previous frames is considered for inference at time t. However, instead of just using feed-forward propagation, we can incorporate both forward and backward propagation when performing inference. For example, the backward passes may be useful if there is strong evidence in the future that a particular hair shape is incorrect. This information can then be propagated to earlier frames and lead to a better overall labeling, at the cost of complicating the inference.

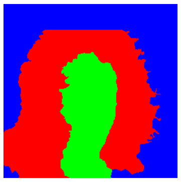
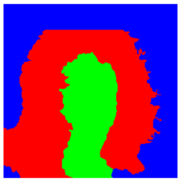
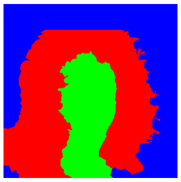
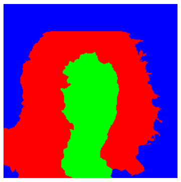
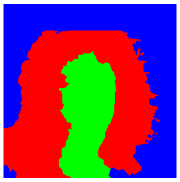
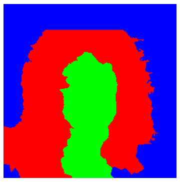
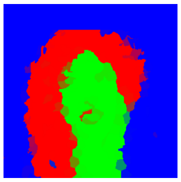
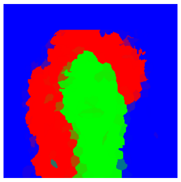
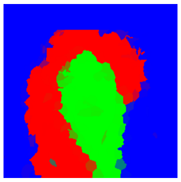
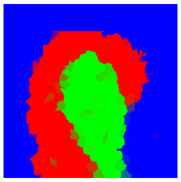
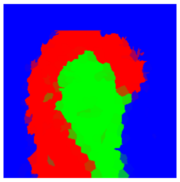
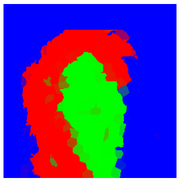
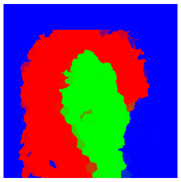
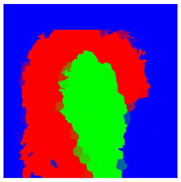
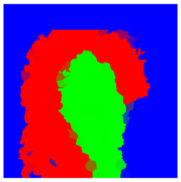
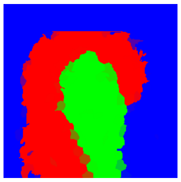
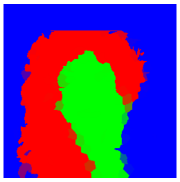
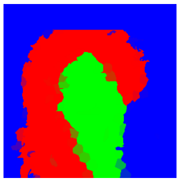
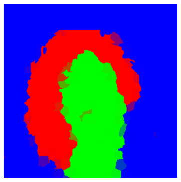
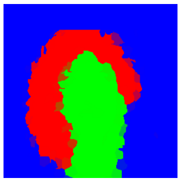
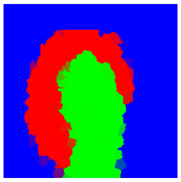
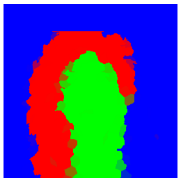
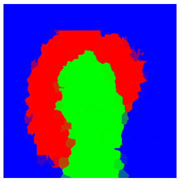
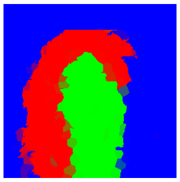
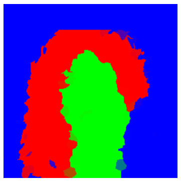
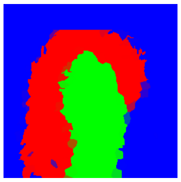
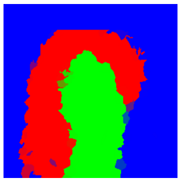
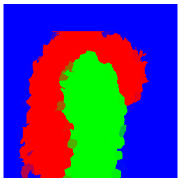
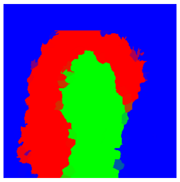
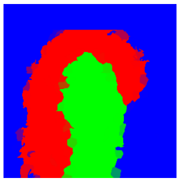
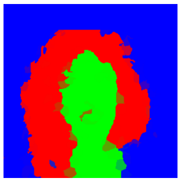
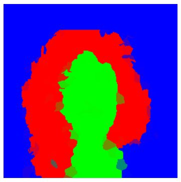
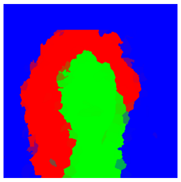
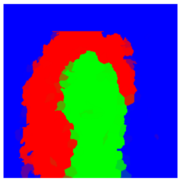
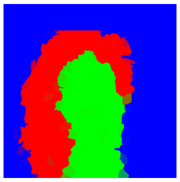
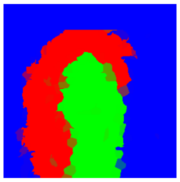
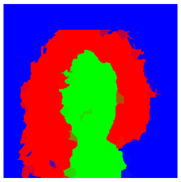
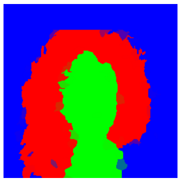
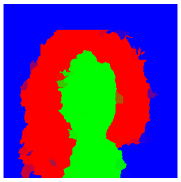
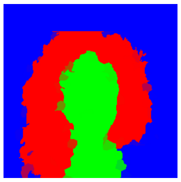
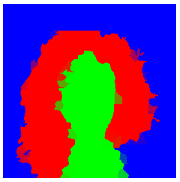
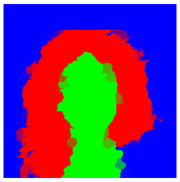
Model	t	t+2	t+4	t+6	t+8	t+10
Original						
Ground Truth						
SCRF						
SCRF + Temporal						
SCRF + RBM						
SCRF + RBM + Temporal						
SCRF + CRBM						
STRF						

Table 5.6. Successful Case. Many of the models had noticeable difficulty labeling the right side of the hair, but the STRF model successfully labeled most of the hair shape.

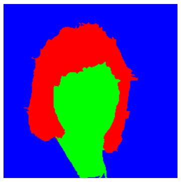
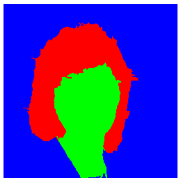
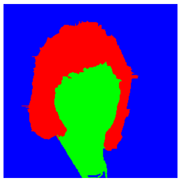
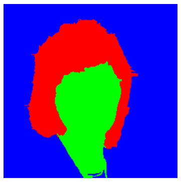
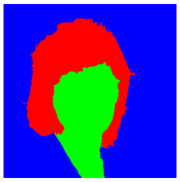
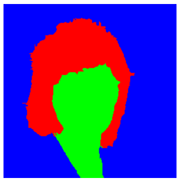
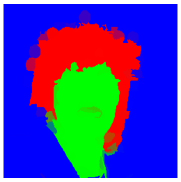
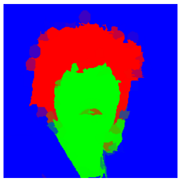
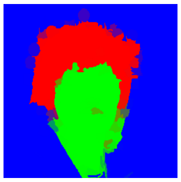
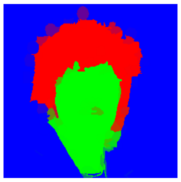
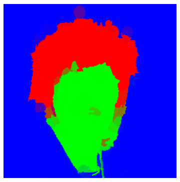
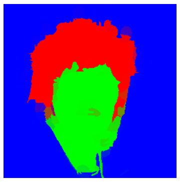
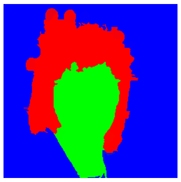
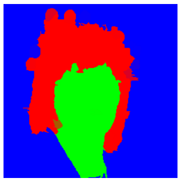
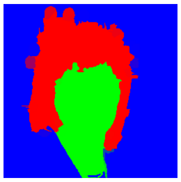
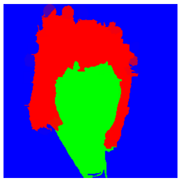
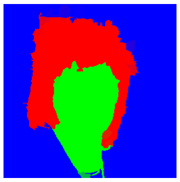
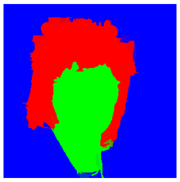
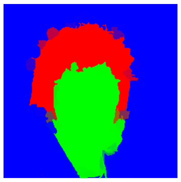
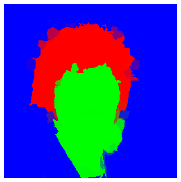
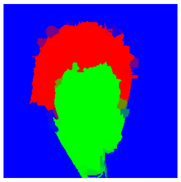
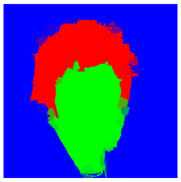
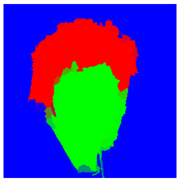
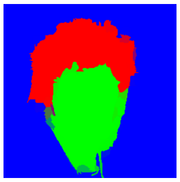
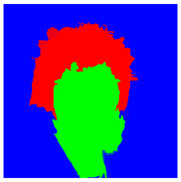
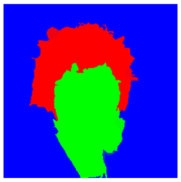
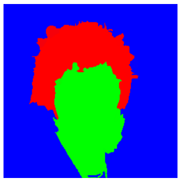
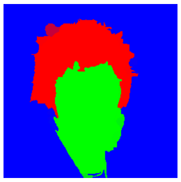
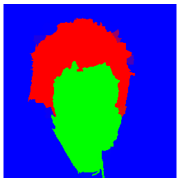
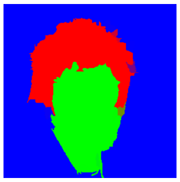
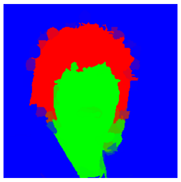
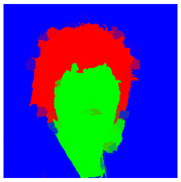
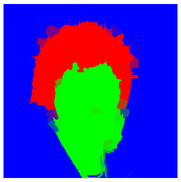
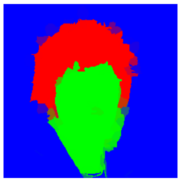
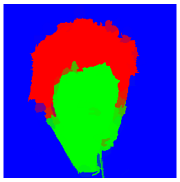
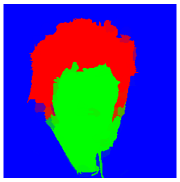
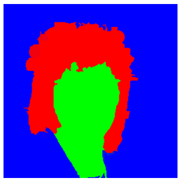
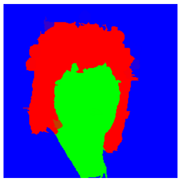
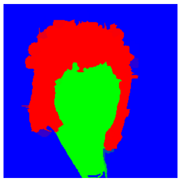
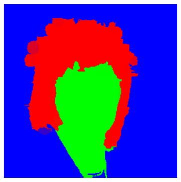
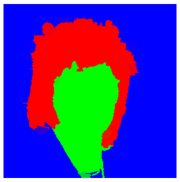
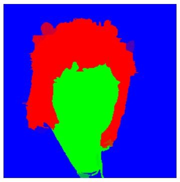
Model	t	t+2	t+4	t+6	t+8	t+10
Original						
Ground Truth						
SCRF						
SCRF + Temporal						
SCRF + RBM						
SCRF + RBM + Temporal						
SCRF + CRBM						
STRF						

Table 5.7. Successful Case. Models with hidden units (bottom four rows) tend to result in a cleaner label shape than models without hidden units. The STRF and SCRF+Temporal are the only models that successfully label the hair on both sides of the woman's face. However, the SCRF+Temporal labeling has an irregular hair shape in frames t through $t + 6$, possibly due to lack of a shape prior.

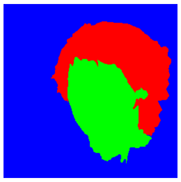
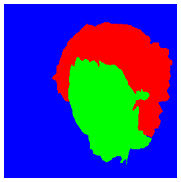
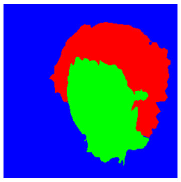
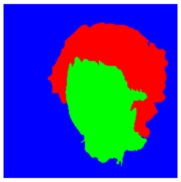
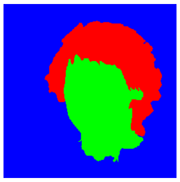
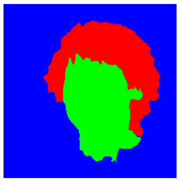
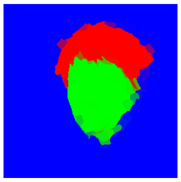
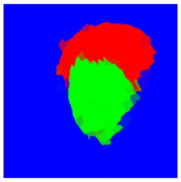
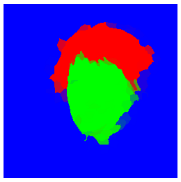
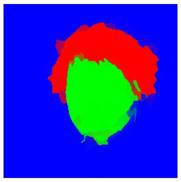
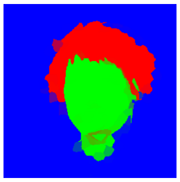
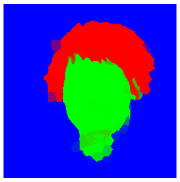
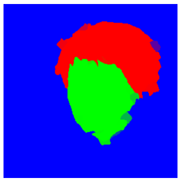
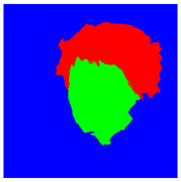
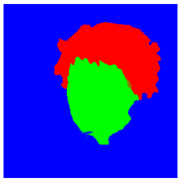
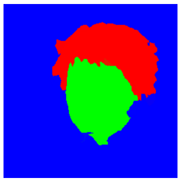
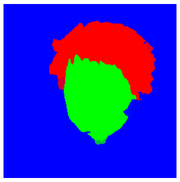
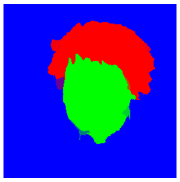
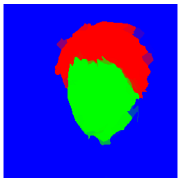
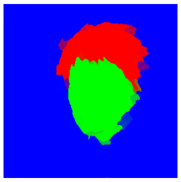
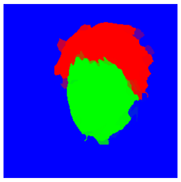
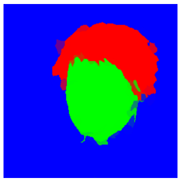
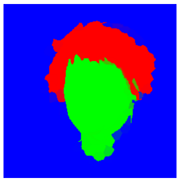
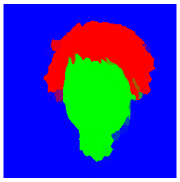
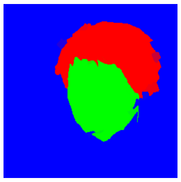
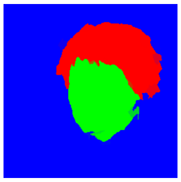
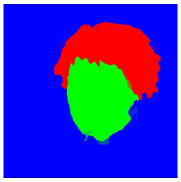
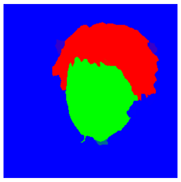
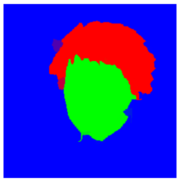
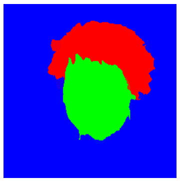
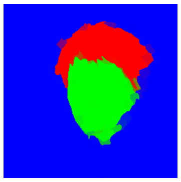
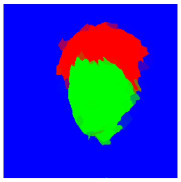
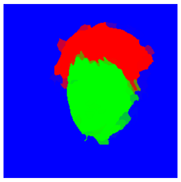
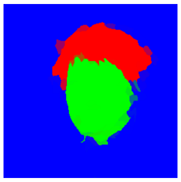
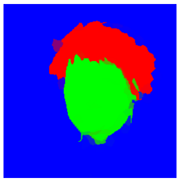
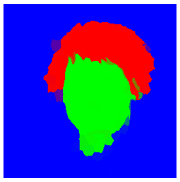
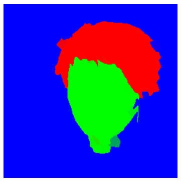
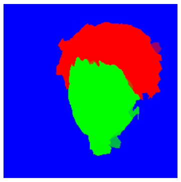
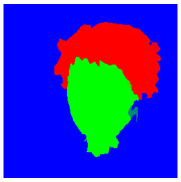
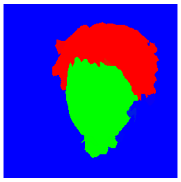
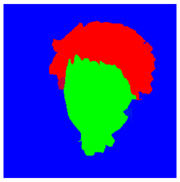
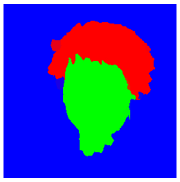
Model	**t**	**t+2**	**t+4**	**t+6**	**t+8**	**t+10**
Original						
Ground Truth						
SCRF						
SCRF + Temporal						
SCRF + RBM						
SCRF + RBM + Temporal						
SCRF + CRBM						
STRF						

Table 5.8. Successful Case. Many of the models generate a good labeling. However, only the STRF model consistently labels the skin region around the neck.

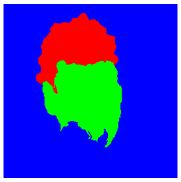
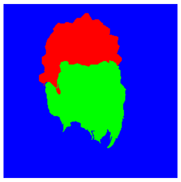
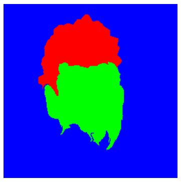
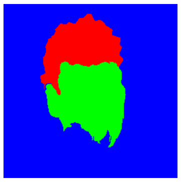
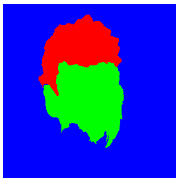
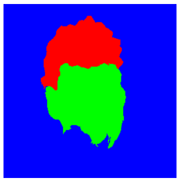
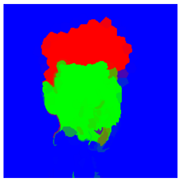
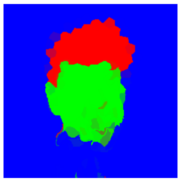
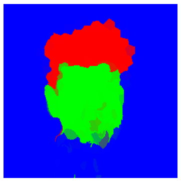
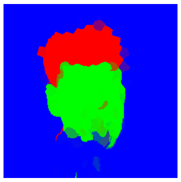
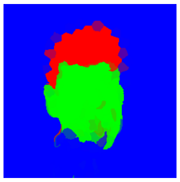
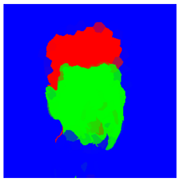
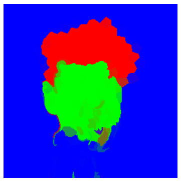
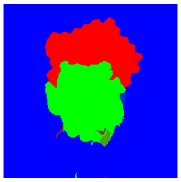
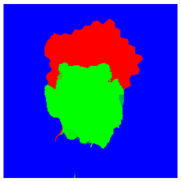
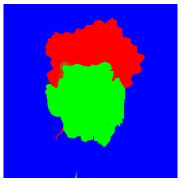
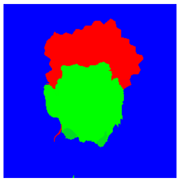
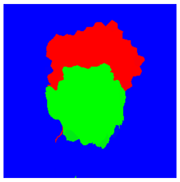
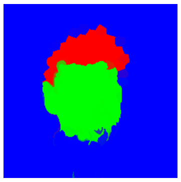
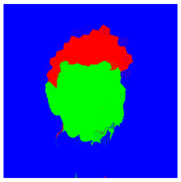
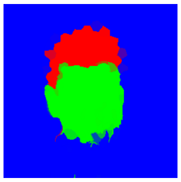
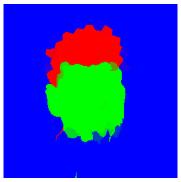
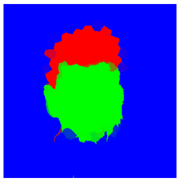
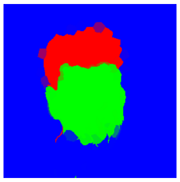
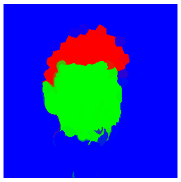
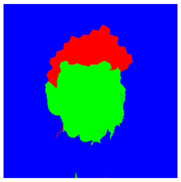
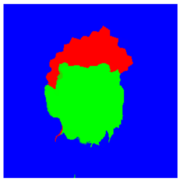
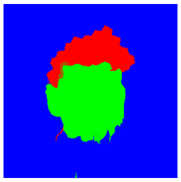
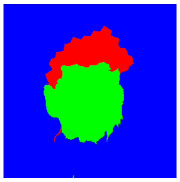
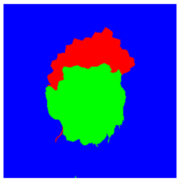
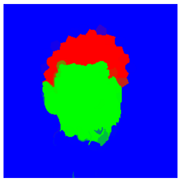
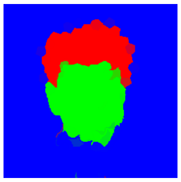
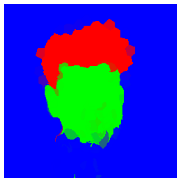
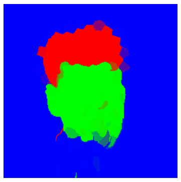
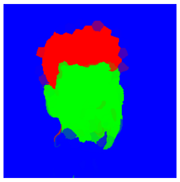
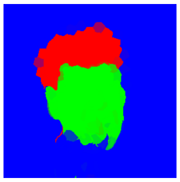
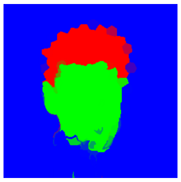
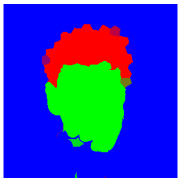
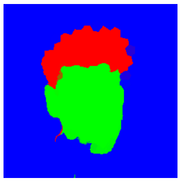
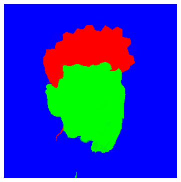
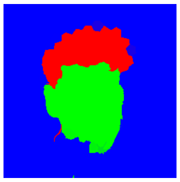
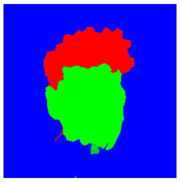
Model	t	t+2	t+4	t+6	t+8	t+10
Original						
Ground Truth						
SCRF						
SCRF + Temporal						
SCRF + RBM						
SCRF + RBM + Temporal						
SCRF + CRBM						
STRF						

Table 5.9. Failure Case. Models with temporal potentials incorrectly guess the wrong hair shape and this error may be propagated through time.

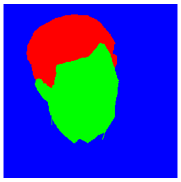
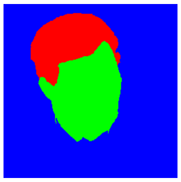
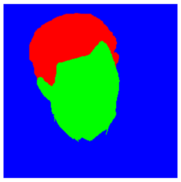
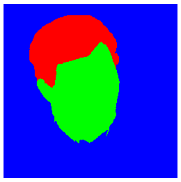
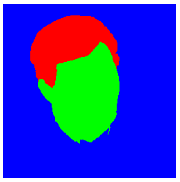
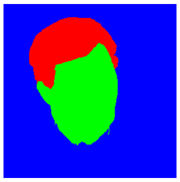
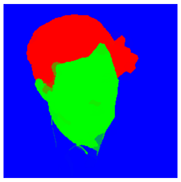
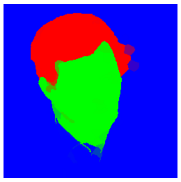
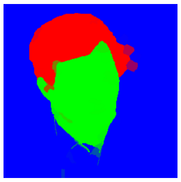
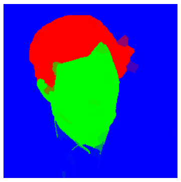
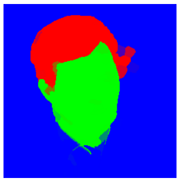
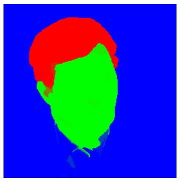
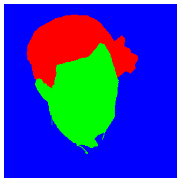
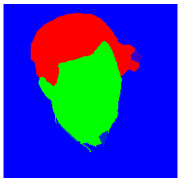
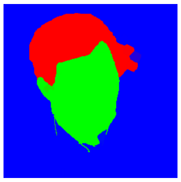
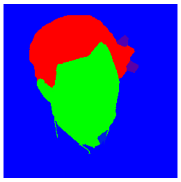
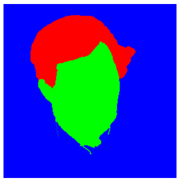
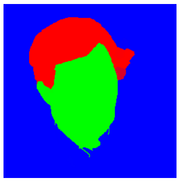
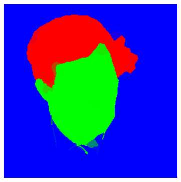
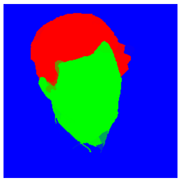
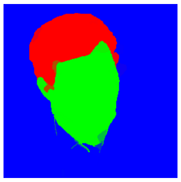
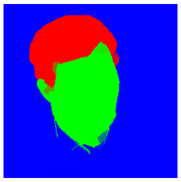
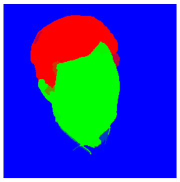
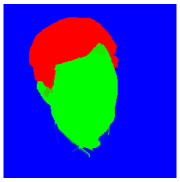
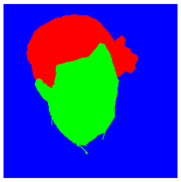
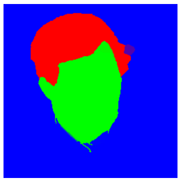
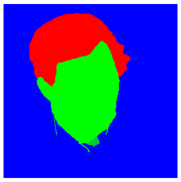
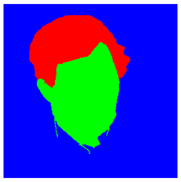
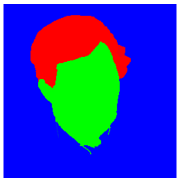
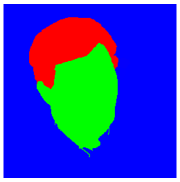
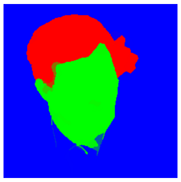
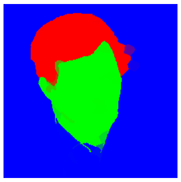
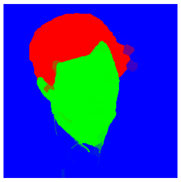
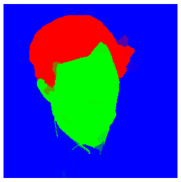
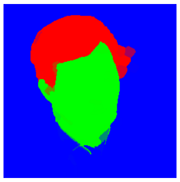
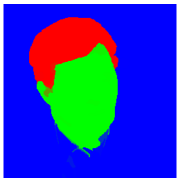
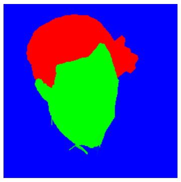
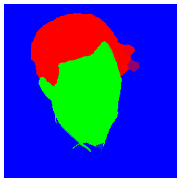
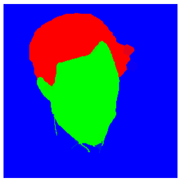
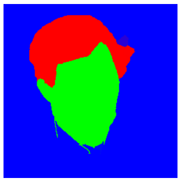
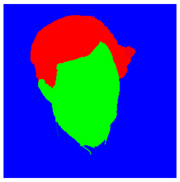
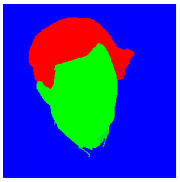
Model	**t**	**t+2**	**t+4**	**t+6**	**t+8**	**t+10**
Original						
Ground Truth						
SCRF						
SCRF + Temporal						
SCRF + RBM						
SCRF + RBM + Temporal						
SCRF + CRBM						
STRF						

Table 5.10. Failure Case. Models with temporal potentials tend to produce an irregular hair shape, possibly because this error is propagated through time.

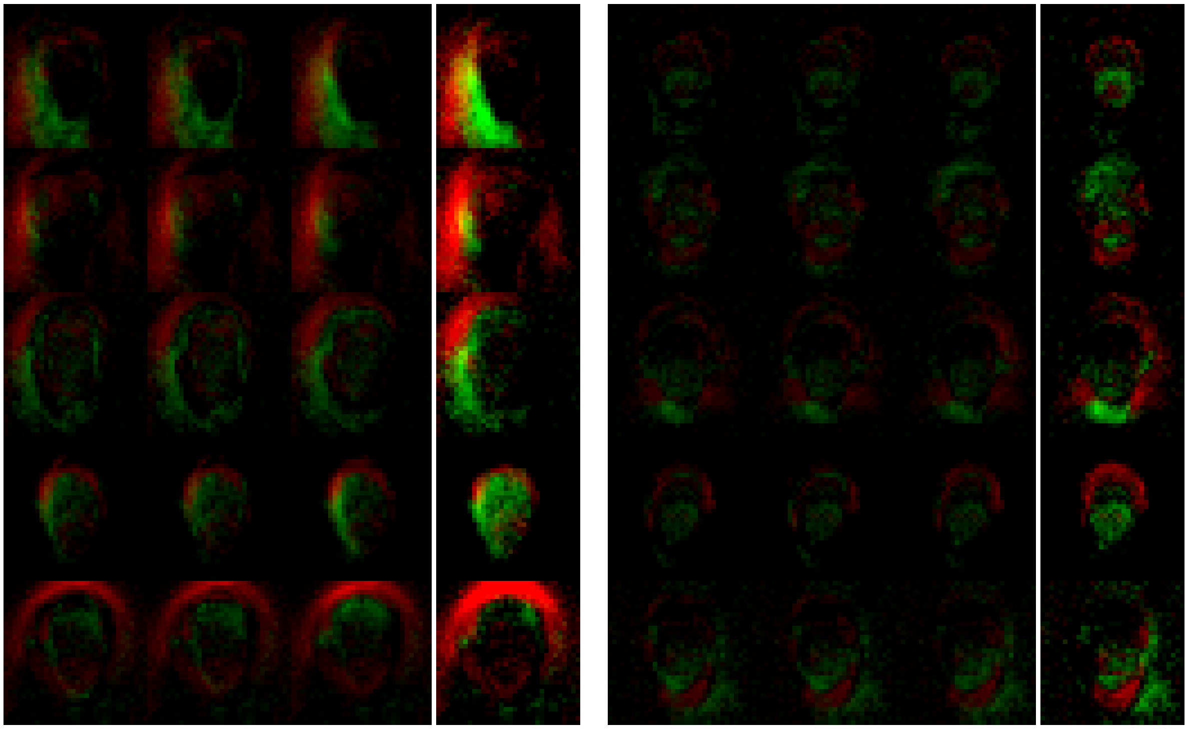
5.6 Learned Filters

This section discusses the learned weights in the CRBM component of the model. As shown in Figure 5.2, the hidden units in the CRBM are connected to both the current visible units (through pairwise weights) and to a history of the virtual visible units (through history weights). The pairwise weights W are shown as green and history weights B are shown as blue.

Figure 5.5 shows samples of 10 different hidden units weight connections to the virtual visible units (these connections are also called filters). These learned weights correspond to the B and W weights from the model, respectively. Note that in Figure 5.2 there are two previous time steps used as history, but the filters in Figure 5.5 show three previous time steps . The history weights are shown to the left of the white line while the pairwise weights are shown to the right of the while line. Each row corresponds to the B, W weights of a particular hidden unit in the CRBM.

In some cases, the history weights seem to learn some of the pose and overall label shape of the corresponding pairwise weights. For example, the filters on the first three rows on the left of Figure 5.5 may be interpreted as a head turning toward the right. For the filters in the right of Figure 5.5, the history weights are similar in appearance to the pairwise weights, suggesting that in this case, the history weights may act as a bias for the label shape. That is, if the CRBM has seen a bearded face in previous frames, it may also expect to see a bearded face in the current frame. In general, the learned history weights B weights seem to be smaller in magnitude than the pairwise weights W.

From the results in Table 5.5, the CRBM seems to help improve classification performance over an RBM, suggesting that the CRBM learns temporal dependencies that are useful for labeling. In addition, in Chapter 4, the hidden unit filters from the RBM in the GLOC model corresponded to semantic attributes and so it would



(a) CRBM filters

(b) CRBM filters

Figure 5.5. Sample of history weights B and pairwise weights W. Each row corresponds to the B, W weights of a particular hidden unit in the CRBM (note that the history in this case uses three previous time steps). The strength of hair labels is shown in **red**, the strength of skin labels is shown in **green**, and the strength of background labels is set to 0 (or black) by default. The "history" weights (B) are shown to the left of the white line in both cases and the corresponding pairwise filters (W) are shown to the right of the white line in both cases. In some cases, the history weights seem to learn some of the pose and overall label shape of the corresponding pairwise weights.

be interesting future work to see if filters from the CRBM can also be interpreted as attributes.

5.7 Discussion

This chapter has introduced a new model called the Shape-Time Random Field (STRF), which incorporates a temporal shape prior (in the form of a CRBM) for use in semantic labeling in face videos. This model builds on previous models to obtain a local, shape, and temporal consistency. We have demonstrated the improved performance of the STRF model both qualitatively and quantitatively over baseline approaches for the semantic labeling of face videos into hair, skin, and background regions.

CHAPTER 6
CONCLUSION

This thesis presented approaches to incorporate Boltzmann machine priors into a discriminative CRF framework for use in the semantic labeling of images and videos. The priors took the form of an RBM for modeling object shape in images and a CRBM for modeling both shape and temporal dependencies in videos. In particular, we presented the GLOC model in Chapter 4 for semantic labeling in images and the STRF model in Chapter 5 for semantic labeling in videos. In both cases, these models demonstrated both quantitative and qualitative improvements over baseline models. In addition, we presented efficient inference and learning algorithms for both models. The data[2] and code [1] for the GLOC model has already been publicly released. In addition, the data and code for the STRF model will be released upon publication.

We focused on the task of semantic labeling of faces into hair, skin, and background regions in images and videos. This particular task is important because of potential applications to surveillance, face verification, and attribute generation. More generally, semantic labeling is useful because it can tell us important information about objects in a scene, their parts, and their context, allowing us to better understand what is *going on* in a scene.

We developed models that can be used to segment and label part regions within complex, real-world face scenes. In particular, our models learned useful information about global shape priors and temporal dependencies which proved useful in improving labeling performance. We have successfully demonstrated the utility of our models for the more constrained problem of hair, skin, and background labeling in face scenes. It is possible that our models can be applied to more general scenes than faces, such as outdoor scenes. However, the labeling task for general scenes is typically less constrained than for face scenes and so our models would have to account for additional complications such as multiple objects (e.g. object classes such as ground, water, or sky) that can appear in varying locations within a scene.

6.1 Future Work

Future work may include the following tasks:

- **Real-time inference for videos.** It may be possible to speed up the STRF inference so that it can be used in real-time. This would make the model more appropriate for use in surveillance applications.

- **Factorize the STRF model.** Currently, the STRF model uses a single set of hidden weights to model both object shape and temporal dependencies. It may be beneficial to separate these behaviors in order to simplify learning in the model and obtain more meaningful, easily interpretable filters. Taylor et al. [85] extended the CRBM for this kind of factorization.

- **Joint training of the STRF model.** The training in the STRF model was done in a piecewise fashion where the temporal SCRF and CRBM components were trained separately and then combined using a single, scalar λ parameter. While this piecewise STRF model outperformed baselines, it is reasonable to expect that a fully jointly trained model may perform even better. This was the case with the GLOC model from Chapter 4.

- **General Scenes.** It may be possible to extend our work from face scenes towards more general scenes. However, as noted earlier, there may be additional complications since general scenes are typically less constrained than face scenes.

BIBLIOGRAPHY

[1] vis-www.cs.umass.edu/GLOC/.

[2] vis-www.cs.umass.edu/lfw/part_labels/.

[3] www.di.ens.fr/~mschmidt/Software/minFunc.html.

[4] A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV ,European Conference on Computer Vision* (2008).

[5] Achanta, Radhakrishna, Shaji, Appu, Smith, Kevin, Lucchi, Aurlien, Fua, Pascal, and Ssstrunk, Sabine. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* (2012), 2274–2282.

[6] Arbeláez, P., Hariharan, B., Gu, C., Gupta, S., Bourdev, L., and Malik, J. Semantic segmentation using regions and parts. In *CVPR* (2012).

[7] Badrinarayanan, Vijay, Galasso, Fabio, and Cipolla, Roberto. Label propagation in video sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition* (2010).

[8] Borenstein, E., Sharon, E., and Ullman, S. Combining top-down and bottom-up segmentation. In *CVPR Workshop on Perceptual Organization in Computer Vision* (2004).

[9] Boykov, Yuri, and Jolly, Marie-Pierre. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV* (2001).

[10] Burl, Michael C., Weber, Markus, and Perona, Pietro. A probabilistic approach to object recognition using local photometry and global geometry, 1998.

[11] Chai, D., and Ngan, K. N. Face segmentation using skin-color map in videophone applications. *IEEE Trans. Cir. and Sys. for Video Technol. 9*, 4 (1999).

[12] Chang, Jason, Wei, Donglai, and III, John W. Fisher. A Video Representation Using Temporal Superpixels. In *CVPR* (2013).

[13] Chen, Liang-Chieh, Papandreou, George, and Yuille, Alan L. Learning a dictionary of shape epitomes with applications to image labeling.

[14] Comaniciu, Dorin, and Meer, Peter. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* (2002).

[15] Comaniciu, Dorin, Ramesh, Visvanathan, and Meer, Peter. Real-time tracking of non-rigid objects using mean shift. pp. 142–149.

[16] Cootes, Timothy F., Edwards, Gareth J., and Taylor, Christopher J. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.* (2001).

[17] Cootes, Timothy F., Taylor, Christopher J., Cooper, David H., and Graham, Jim. Active shape models-their training and application. *Computer Vision and Image Understanding* (1995).

[18] Dalal, Navneet, and Triggs, Bill. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition* (June 2005), vol. 2, pp. 886–893.

[19] Dementhon, Daniel. Spatio-temporal segmentation of video by hierarchical mean shift analysis. In *Center for Automat. Res., U. of Md, College Park* (2002).

[20] Eslami, S. M. Ali, Heess, Nicolas, and Winn, John. The shape Boltzmann machine: A strong model of object shape. In *CVPR* (2012).

[21] Eslami, S. M. Ali, and Williams, Christopher K. I. A generative model for parts-based object segmentation. In *NIPS* (2012).

[22] Felzenszwalb, Pedro F., Girshick, Ross B., McAllester, David, and Ramanan, Deva. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* (2010).

[23] Felzenszwalb, Pedro F., and Huttenlocher, Daniel P. Pictorial structures for object recognition. *International Journal of Computer Vision* (2005).

[24] Felzenszwalb, Pedro F., McAllester, David A., and Ramanan, Deva. A discriminatively trained, multiscale, deformable part model. In *CVPR* (2008).

[25] Fergus, R., Perona, P., and Zisserman, A. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition* (2003), vol. 2, pp. 264–271.

[26] Fischler, M. A., and Elschlager, R. A. The representation and matching of pictorial structures. *IEEE Trans. Comput.* (1973).

[27] Floros, Georgios. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *CVPR* (2012).

[28] Fukunaga, K., and Hostetler, L. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inf. Theor.* (2006).

[29] Gavrila, Dariu M. A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Trans. Pattern Anal. Mach. Intell.* (2007).

[30] Geman, Stuart, and Geman, Donald. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* (1984).

[31] Gould, Stephen, Fulton, Richard, and Koller, Daphne. Decomposing a scene into geometric and semantically consistent regions. In *ICCV* (2009).

[32] Grundmann, M., Kwatra, V., Han, M., and Essa, I. Efficient hierarchical graph based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[33] Hanson, A. R., and Riseman, E. M. VISIONS: A computer system for interpreting scenes. In *Computer Vision Systems*. Academic Press, 1978.

[34] Haralick, Robert M., and Shapiro, Linda G. Image segmentation techniques.

[35] He, X., Zemel, R., and Ray, D. Learning and incorporating top-down cues in image segmentation. In *ECCV* (2006).

[36] He, X., Zemel, R.S., and Carreira-Perpinán, M.A. Multiscale conditional random fields for image labeling. In *CVPR* (2004).

[37] Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation 14*, 8 (2002), 1771–1800.

[38] Hinton, G. E., and Sejnowski, T. J. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. MIT Press, Cambridge, MA, USA, 1986, ch. Learning and Relearning in Boltzmann Machines.

[39] Hinton, Geoffrey E., and Osindero, Simon. A fast learning algorithm for deep belief nets. *Neural Computation* (2006).

[40] Horowitz, Steven L., and Pavlidis, Theodosios. Picture segmentation by a tree traversal algorithm. *J. ACM* (1976).

[41] Huan, Gamg. personal communication.

[42] Huang, G. B., Lee, H., and Learned-Miller, E. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR* (2012).

[43] Huang, Gary B., Jain, Vidit, and Learned-Miller, Erik. Unsupervised joint alignment of complex images. In *ICCV* (2007).

[44] Huang, Gary B., Mattar, Marwan, Lee, Honglak, and Learned-Miller, Erik. Learning to align from scratch. In *NIPS* (2012).

[45] Huang, Gary B., Narayana, Manjunath, and Learned-Miller, Erik. Towards unconstrained face recognition. In *CVPR Workshop on Perceptual Organization in Computer Vision* (2008).

[46] Huang, Gary B., Ramesh, Manu, Berg, Tamara, and Learned-Miller, Erik. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, 2007.

[47] Kae, Andrew, Sohn, Kihyuk, Lee, Honglak, and Learned-Miller, Erik. Augmenting crfs with boltzmann machine shape priors for image labeling. In *CVPR* (2013).

[48] Kalman, Rudolph Emil. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D (1960), 35–45.

[49] Kohli, Pushmeet, Ladický, L'Ubor, and Torr, Philip H. Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vision* (2009).

[50] Kumar, M. P., Torr, P. H. S., and Zisserman, A. OBJ CUT. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego* (2005).

[51] Kumar, Neeraj, Berg, Alexander C., Belhumeur, Peter N., and Nayar, Shree K. Attribute and simile classifiers for face verification. In *ICCV* (2009).

[52] Lafferty, J., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML* (2001).

[53] Le, Q. V., Ranzato, M. A., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. Building high-level features using large scale unsupervised learning. In *ICML* (2012).

[54] LeCun, Yann, Chopra, Sumit, Hadsell, Raia, Ranzato, Marc'Aurelio, and Huang, Fu-Jie. A tutorial on energy-based learning. In *Predicting Structured Data* (2006), G. Bakir, T. Hofman, B. Schölkopf, A. Smola, and B. Taskar, Eds., MIT Press.

[55] Lee, K., Anguelov, D., Sumengen, B., and Gokturk, S.B. Markov random field models for hair and face segmentation. In *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on* (2008).

[56] Li, Haoxiang, Hua, Gang, Lin, Zhe, Brandt, Jonathan, and Yang, Jianchao. Probabilistic elastic matching for pose variant face verification.

[57] Li, Hongliang, and Ngan, King N. Saliency model-based face segmentation and tracking in head-and-shoulder video sequences. *J. Vis. Comun. Image Represent.* (2008).

[58] Li, Hongliang, Ngan, King Ngi, and Liu, Qiang. Faceseg: Automatic face segmentation for real-time video. *IEEE Transactions on Multimedia* (2009).

[59] Li, S. Z. Markov random field models in computer vision, 1994.

[60] Li, Yujia, Tarlow, Daniel, and Zemel, Richard. Exploring compositional high order pattern potentials for structured output learning. In *CVPR* (2013).

[61] Lievin, M., and Luthon, F. Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video. *Trans. Img. Proc.* (2004).

[62] Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Hierarchical face parsing via deep learning. In *CVPR* (2012), IEEE.

[63] Malik, J., Belongie, S., Shi, J., and Leung, T. Textons, contours and regions: Cue integration in image segmentation. In *ICCV* (1999).

[64] Martin, D., Fowlkes, C., and Malik, J. Learning to detect natural image boundaries using brightness and texture. In *NIPS* (2002).

[65] Mnih, Volodymyr, Larochelle, Hugo, and Hinton, Geoffrey. Conditional restricted boltzmann machines for structured output prediction. In *UAI* (2011).

[66] Mori, G. Guiding model search using segmentation. In *Proc. 10th Int. Conf. Computer Vision* (2005), vol. 2, pp. 1417–1423.

[67] Morris, R. D., Descombes, X., and Zerubia, J. The ising/potts model is not well suited to segmentation tasks. *IEEE DSP Workshop,* (1996.).

[68] Otsu, N. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics* (1979).

[69] Porter, Thomas, and Duff, Tom. Compositing digital images. *SIGGRAPH Comput. Graph.* (1984).

[70] Rastegari, M., Farhadi, A., and Forsyth, D. Attribute discovery via predictable discriminative binary codes. In *ECCV* (2012).

[71] Ren, Xiaofeng, and Malik, Jitendra. Learning a classification model for segmentation. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2* (2003), ICCV '03.

[72] Rother, Carsten, Kolmogorov, Vladimir, and Blake, Andrew. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH 2004 Papers* (2004), SIGGRAPH '04.

[73] Russell, Bryan C., Torralba, Antonio, Murphy, Kevin P., and Freeman, William T. Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision* (2008).

[74] Salakhutdinov, Ruslan, and Hinton, Geoffrey. Deep Boltzmann machines. In *AISTATS* (2009).

[75] Sato, Kengo, and Sakakibara, Yasubumi. Rna secondary structural alignment with conditional random fields. In *ECCB/JBI* (2005).

[76] Saul, L.K., Jaakkola, T., and Jordan, M.I. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research 4* (1996), 61–76.

[77] Scheffler, C., Odobez, J.M., and Marconi, R. Joint adaptive colour modelling and skin, hair and clothing segmentation using coherent probabilistic index maps. In *BMVC* (2011).

[78] Sha, Fei, and Pereira, Fernando. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* (2003), NAACL '03.

[79] Shi, Jianbo, and Malik, Jitendra. Normalized cuts and image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)* (1997), CVPR '97.

[80] Shotton, Jamie, Winn, John, Rother, Carsten, and Criminisi, Antonio. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision* (2009).

[81] Sminchisescu, Cristian, Kanaujia, Atul, and Metaxas, Dimitris. Conditional models for contextual human motion recognition. *Comput. Vis. Image Underst.* (2006).

[82] Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory. *Parallel distributed processing: Explorations in the microstructure of cognition 1* (1986), 194–281.

[83] Stauffer, Chris, and Grimson, W. Eric L. Adaptive background mixture models for real-time tracking. In *CVPR'99* (1999), pp. 2246–2252.

[84] Sutton, Charles, and Mccallum, Andrew. *Introduction to Conditional Random Fields for Relational Learning.* 2006.

[85] Taylor, Graham W., and Hinton, Geoffrey E. Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ICML '09.

[86] Taylor, Graham W., Hinton, Geoffrey E., and Roweis, Sam. Modeling human motion using binary latent variables. In *NIPS* (2006).

[87] Vazquez-Reina, Amelio, Avidan, Shai, Pfister, Hanspeter, and Miller, Eric. Multiple hypothesis video segmentation from superpixel flows. In *Proceedings of the 11th European Conference on Computer Vision: Part V* (2010), ECCV'10.

[88] Viola, Paul, and Jones, Michael. Robust real-time face detection. *International Journal of Computer Vision* (2004).

[89] Wang, N., Ai, H., and Lao, S. A compositional exemplar-based model for hair segmentation. In *ACCV* (2011).

[90] Wang, Nan, Ai, Haizhou, and Tang, Feng. What are good parts for hair shape modeling? In *CVPR* (2012).

[91] Wang, Sy Bor, Quattoni, Ariadna, Morency, Louis-Philippe, and Demirdjian, David. Hidden conditional random fields for gesture recognition. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2* (2006), CVPR '06.

[92] Wang, Yang, and Ji, Qiang. A dynamic conditional random field model for object segmentation in image sequences. In *CVPR* (2005).

[93] Warrell, Jonathan, and Prince, Simon J. D. Labelfaces: Parsing facial features by multiclass labeling with an epitome prior. In *Proceedings of the 16th IEEE International Conference on Image Processing* (2009), ICIP'09.

[94] Weber, Markus, Welling, Max, and Perona, Pietro. Unsupervised learning of models for recognition. In *Proceedings of the 6th European Conference on Computer Vision-Part I* (2000).

[95] Wertheimer, Max. Laws of organization in perceptual forms. *A Sourcebook of Gestalt Psycychology* (1938).

[96] Winn, J., and Jojic, N. Locus: Learning object classes with unsupervised segmentation. In *ICCV* (2005), vol. 1, IEEE, pp. 756–763.

[97] Wolf, Lior, Hassner, Tal, and Maoz, Itay. Face recognition in unconstrained videos with matched background similarity. In *CVPR* (2011).

[98] Wolf, Lior, Hassner, Tal, and Taigman, Yaniv. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *IEEE-TPAMI 33*, 10 (2011), 1978–1990.

[99] Xu, C., and Corso, J. J. Evaluation of super-voxel methods for early video processing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (2012).

[100] Xu, C., Xiong, C., and Corso, J. J. Streaming hierarchical video segmentation. In *Proceedings of European Conference on Computer Vision* (2012).

[101] Yacoob, Y., and Davis, L.S. Detection and analysis of hair. *IEEE-PAMI 28*, 7 (2006), 1164–1169.

[102] Yakimovsky, Yoram, and Feldman, Jerome A. A semantics-based decision theory region analyzer. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence* (1973), IJCAI'73.

[103] Yang, Yi, and Ramanan, D. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (2011), CVPR '11.