IJMTST is UGC Approved Journal with Journal Id: 43137

International Journal for Modern Trends in Science and Technology, 8(S08):10-18, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST08S0802
Available online at: https://www.ijmtst.com/vol8si08.html

Convolutional Neural Network based on Image Segmantation

Shaik. RaziyaSultana | K.Tejaswi | M.Pushpa Latha

Computer Science and Engineering , Chalapathi Institute of Engineering and Technology, Lam, Guntur, A.P, India

To Cite this Article
Shaik. RaziyaSultana, K.Tejaswi and M.Pushpa Latha. Convolutional Neural Network based on Image Segmantation. International Journal for Modern Trends in Science and Technology 2022, 8(S08), pp. 10-18. https://doi.org/10.46501/IJMTST08S0802

Article Info
Received: 26 May 2022; Accepted: 24 June 2022; Published: 28 June 2022.

ABSTRACT

By modeling long-range interactions, dense CRFs provide a more detailed labeling compared to their sparse counterparts. Variational inference in these dense models is performed using a ltering based mean field algorithm in order to obtain a fully factorized distribution minimizing the Kull back Leibler divergence to the true distribution. In contrast to the continuous relaxation based energy minimisation algorithms used for sparse CRFs, the mean field algorithm fails to provide strong theoretical guarantees on the quality of its solutions. we solve a convex quadratic programming (QP) relaxation using the efficient Frank-Wolfe algorithm. This also allows us to solve difference of convex relaxations via the iterative concave convex procedure where each iteration requires solving a convex QP. Finally, we develop a novel divide and conquer method to compute the subgradients of a linear programming relaxation that provides the best theoretical bounds for energy minimisation. We demonstrate the advantage of continuous relaxations over the widely used mean field algorithm on publicly available datasets. crowd density from static images of highly dense crowds. We use a combination of deep and shallow, fully convolutional networks to predict the density map for a given crowd image. Such a combination is used for effectively capturing both the high-level semantic information (face/body detectors) and the low level features (blob detectors), that are necessary for crowd counting under large scale variations. As most crowd datasets have limited training samples (<100 images) and deep learning based approaches require large amounts of training data, we perform multi-scale data augmentation. Augmenting the training samples in such a manner helps in guiding the CNN to learn scale invariant representations..

KEYWORDS:*Energy minimisation, Dense CRF, Interference, linear Programming, Quadratic Programming.*

INTRODUCTION

Crowd counting is a crucial component of such an automated crowd analysis system. This involves estimating the number of people in the crowd, as well as the distribution of the crowd density over the entire area of the gathering. Identifying regions with crowd density above the safety limit can help in issuing prior warnings and can prevent potential crowd crushes. Estimating the crowd count also helps in quantifying the significance of the event and better handling of logistics and infrastructure for the gathering. Traditionally, computer vision methods have employed sparse connectivity structures, such as 4 or 8 connected grid CRFs. Their popularity leads to a considerable research e ort in

efficient energy minimization algorithms. One of the biggest successes of this e ort was the development of several accurate continuous relaxations of the underlying discrete optimization problem. An important advantage of such relaxations is that they lend themselves easily to analysis, which allows us to compare them theoretically, as well as establish bounds on the quality of their solutions. Recently, the in uential work of Krahenbuhl and Koltun has popularized the use of dense CRFs, where each pair of random variables is connected by an edge. Dense CRFs capture useful long-range interactions thereby providing near details on the labeling.

While the mean field algorithm does not provide any theoretical guarantees on the energy of the solutions, the use of a richer model, namely dense CRFs, still allows us to obtain a significant improvement in the accuracy of several computer vision applications compared to sparse CRFs. However, this still leaves open the intriguing possibility that the same altering approach that enabled the efficient mean field algorithm can also be used to speed up energy minimization algorithms based on continuous relaxations. In this work, we show that this is indeed possible.

RELATED WORKS:

Some works in the crowd counting literature experiment on datasets having sparse crowd scenes, such as UCSD dataset , Mall dataset and PETS dataset . In contrast, our method has been evaluated on highly dense crowd images which pose the challenges discussed in the previous section. Methods introduced in and exploit patterns of motion to estimate the count of moving objects. How-ever, these methods rely on motion information which can be obtained only in the case of continuous video streams with a good frame rate, and do not extend to still image crowd counting.

Krahenbuhl and Koltun popularized the use of densely connected CRFs at the pixel level, resulting in significant improvements both in terms of the quantitative performance and in terms of the visual quality of their results. By restricting themselves to Gaussian edge potentials, they made the computation of the message in parallel mean field feasible. This was achieved by formulating message computation as a convolution in a higher dimensional space, which enabled the use of an efficient filter based method.

While the original workused a version of mean field that is not guaranteed to converge, their follow up paper proposed a convergent mean field algorithm for negative semi definite label compatibility functions. Recently, Baque et al. Presented a new algorithm that has convergence guarantees in the general case. Vineet et al. extended the mean field model to allow the addition of higher orderterms on top of the dense pairwise potentials, enabling the use of co-occurrence potentials and P ^{n}-Potts models. The success of the inferencealgorithms naturally lead to research in learning the parameters of dense CRFs. Combining them with Fully Convolution Neural Networks has resulted in high performance on semantic segmentation applications [16]. Several works showed independently how to jointly learn the parameters of the unary and pair wise potentials of the CRF. These methods led to significant improvements on various computer vision applications, by increasing the quality of the energy function to be minimized by mean field.

In this paper, we use the same filter based method as the one employed in mean- field. We build on it to solve continuous relaxations of the original problem that have both convergence and quality guarantees. Our work can be viewed as a complementary direction to previous research trends in dense CRFs. While improved mean field and learnt the parameters, we focus on the energy minimization problem.

PRELIMINARIES

Before describing our methods for energy minimisation on dense CRF, we establish the necessary notation and background information.

Dense CRF Energy Function: we define a dense CRF on a set of N random variables $_{\chi} = \{X_1, \ldots . X_N\}$ each of which can take one label fro a set of M labels $\mathcal{L} = \{l_1, \ldots . l_M\}$. To describe a labelling, we use a vector x of size N such that its element x_a is the label taken by the random variable X_a. The energy associated with a given labelling is defined as:

$$E(x) = \sum_{a=1}^{N} \phi_a\,(x_a) + \sum_{a=1}^{N} \sum_{\substack{b=1 \\ b \neq a}}^{N} \psi_{a,b}(x_a, x_b) \quad (1)$$

Here, $\phi_a(x_a)$is called the unary potential for the random variable X_a taking the label x_a. The term $\psi_{a,b}(x_a, x_b)$ is called the pair wise potential for random variable X_a and X_b taking the labels x_a and x_b respectively. The energy

minimisation problem on this CRF can be written as

$$X^* = \underset{x}{argmin} E(x) \qquad (2)$$

Gaussian Pair wise Potential:

Similar to previous work we consider arbitrary unary potential and Gaussian pairwise potentials. Specifically, the form of the pairwise potentials is given by:

$$\Psi_{a,b}(i,j) = \mu(i,j) \sum_m \omega^{(m)} k(f_a^{(m)}, f_b^{(m)}) \qquad (3)$$

$$k(f_a, f_b) = \exp(\frac{-\|f_a - f_b\|^2}{2}) \qquad (4)$$

We refer to the term $\mu(i,j)$ is a label compatibility function between the labels I and j. An example of a label compatibility function is the Potts model, where $\mu_{potts}(i,j) = [i \neq j]$, that is

$\mu_{potts}(i,j) = 1 \ if i \neq j and \ 0 \ otherwise.$

Note that the label compatibility does not depend on the image. The other term, called the pixel compatibility function, is a mixture of Gaussian kernels k(.,.).

The coefficient of the mixture are the weights ω^m. The $f_a^{(m)}$ are the features describing the random variables X_a. Note that the pixel compatibility does not depend on the labeling. In practice, we use the position and RGB values of a pixel as features.

IP Formulation:

We now introduce a formulation of the energy minimization problem that is more amenable to continuous relaxations. Specifically, formulate it as an Integer Program (IP) and then relax it to obtain a continuous optimization problem. To this end, we define the vector y whose components $y_a(i)$ are indicator variables specifying whether or not the random variable X_a takes the label i. using this notation, we can rewrite the energy minimization problem as an IP:

$$\min \sum_{a=1}^N \sum_{i \in L} \phi_a(i) y_a(i) + \sum_{a=1}^N \sum_{\substack{b=1 \\ b \neq a}}^N \sum_{i,j \in L} \psi_{a,b}(i,j), y_a(i) y_b(j), \quad s.t. \sum_{i \in L} y_a(i) = 1 \quad \forall_a \in [1,N], \qquad (5)$$

$$y_a(i) \in \{0,1\} \forall_a \in [1,N] \forall_i \in L.$$

The first set of constraints model the fact each random variable has to be assigned exactly one label. The second set of constraints enforce the optimization variable $y_a(i)$ to be binary. Note that the objective function is equal to the energy of the labeling encoded by **y.**

FILTER- BASED METHOD

A key component of our algorithms is the filter-based method of Adams et al. It computes the following operation:

$$\forall_a \in [1,N], \ v_a^{'} = \sum_{b=1}^N k(f_a, f_b) v_b \qquad (6)$$

Where $v_a^{'}, v_b \in \mathbb{R} \ and \ k(.,.)$ is a Guassian kernel. Performing this operation the naïve way would result in computing a sum on N elements for each of the N terms that we want to compute. The resulting complexity would be $O(N^2)$. The filter- based method allows us to perform it approximately with $O(N)$ complexity. The accuracy of the approximation made by the filter-based method is explored in the future discussion.

Quadratic Programming Relaxation:

The filter-based method can be used to optimize our first continuous relaxation, namely the convex quadratic programming (QP) Relaxation.

Notation: In order to concisely specify the QP relaxation, we require some additional notations. The vector φ contains the unary terms. the matrix μ corresponds to the label compatibility function. The Guassian kernels associated with the m-th features are represented by their Gram matrix $K_{a,b}^{(m)} = k(f_a^{(m)}, f_b^{(m)})$. The kronecher product is denoted by $\otimes$. the matrix Ψ represents the pair wise terms and is defined as follows:

$$\Psi = \mu \otimes (\sum_m K^{(m)} - I_N) \qquad (7)$$

Where I_N is the identity matrix. Under this notation, the IP can be concisely written as

$$\min \phi^T y + y^T \Psi y \ ,$$

$$s.t \ y \in I \qquad (8)$$

With I being the feasible set of integer solution, as defined in equation (5).

Relaxation: In general, IP such as (8) are NP-hard problems. Relaxing the integer constraint on the indicator variables to allow fractional values between 0 and 1 results in the QP formulation. Formally, the feasible set of our minimization problem becomes:

$$\mu = \left\{ y \ such \ taht \ \begin{matrix} \sum_{i \in L} y_a(i) = 1 & \forall_a \in [1,N] \\ y_a(i) \geq 0 & \forall_a \in [1,N], \forall_i \in L \end{matrix} \right\} \qquad (9)$$

However , this QP is still NP-Hard of the objective function is non-convex . To alleviate this difficulty, The QP minimization to the following convex problem

$$\min S_{cvx}(y) = (\phi - d)^T y + y^T (\Psi + D) y,$$

$$s.t \quad y \in \mu, \qquad (10)$$

Where the vector **d** is defined as follows

$$d_a(i) = \sum_{\substack{b=1 \\ b \neq a}}^N \sum_{j \in L} |\psi_{a,b}(i,j)|, \qquad (11)$$

And D is the square diagonal matrix with **d** its diagonal.

Gradient computation

Since the objective function is quadratic, its gradient

can be computed as

$\nabla S_{cvx}(\boldsymbol{y}) = (\phi - d) + 2(\Psi + D)y$ (12)

What makes this equation expensive to compute in a naïve way is the matrix product with Ψ . we observe that this operation can be performed using the filter-based method in linear time. Note that the other matrix- vector product **D**y, is not expensive (linear in N) since D is a diagonal matrix.

Ground truth:

The exact position of the head annotations is often ambiguous, and varies from annotator to annotator (forehead, centre of the face etc.), making CNN training difficult. In the authors have trained a deep network to predict the total crowd count in an image patch. But using such a ground truth would be suboptimal, as it wouldn't help in determining which regions of the image actually con-tribute to the count and by what amount. Zhang et al. have generated ground truth by blurring the binary head annotations, using a kernel that varies with respect to the perspective map of the image. However, generating such perspective maps is a laborious task and involves manually labelling several pedestrians by marking their height. We generate our ground truth by simply blurring each head annotation using a Gaussian kernel normalized to sum to one. This kind of blurring causes the sum of the density map to be the same as the total number of people in the crowd. Preparing the ground truth in such a fashion makes the ground truth easier for the CNN to learn, as the CNN no longer needs to get the exact point of head annotation right. It also provides information on which regions contribute to the count, and by how much. This helps in training the CNN to predict both the crowd density as well as the crowd count correctly.

Data Augmentation:

As CNNs require a large amount of training data, we per-form an extensive augmentation of our training dataset. We primarily perform two types of augmentation. The first type of augmentation helps in tackling the problem of scale verifications in crowd images, while the second type improves the CNN's performance in regions where it is highly susceptible to making mistakes i.e., highly dense crowd regions.

Algorithm 1 Frank-Wolfe Algorithm

1: Get $y^0 \in \mathcal{H}$

2: **while** not converged **do**

3: compute the gradient at y^t as $g = \nabla f(y^t)$

4: compute the conditional gradient as $s = \underset{s \in \mathcal{H}}{argmin} \langle s, g \rangle$

5: compute a step-size $\alpha = argmin_{\alpha \in [0,1]} f(\alpha y^t + (1-\alpha)s)$

6: move towards the negative conditional gradient $y^{t+1} = \alpha y^t + (1-\alpha)s$

7: **endwhile**

Conditional gradient

The conditional gradient is obtained by solving

$\underset{s \in \mathcal{H}}{argmin} (s, \nabla S_{cvx}(\boldsymbol{y})$ (13)

Step Size determination:

In the original Frank-Wolfe algorithm, the step size is simply chosen using line search. However we observe that, in our case, the optimal can be computed by solving a second-order polynomial function of a single variable, which has a closed form solution that can be obtained efficiently. This observation has been previously exploited in the context of Structural SVM. With careful reutilization of computations, this step can be performed without additional filter-based method calls. By choosing the optimal step size at each iteration, we reduce the number of iterations needed to reach convergence.

Difference of Convex Relaxation:

The objective function of a general DC program can be specified as

$S_{CCCP}(y)$=p(y)-q(y) (14)

One can obtain one of its local minima using the Concave-Convex Procedure(CCCP). In order to exploit the CCCP Algorithm for DC programs, we observe that the QP equation (8) can be rewritten as $\underset{y}{min} \phi^T + y^T(\Psi + D)y - y^T Dy$,

$s.t \;\; s \in \mathcal{H}$ (15)

Formally, we can define $p(y) = \phi^T + y^T(\Psi + D)y$ and $q(y) = y^T Dy$, which are both convex in **y.**

Algorithm 2 CCCP algorithm

1: Get $y^0 \in \mathcal{H}$

2: **while** not converged **do**

3: Linearise the concave part $g = \nabla q(y^t)$

4: minimize a convex upper-bound $y^{T+1} = argmin_{y \in \mathcal{H}} p(y) - g^T y$

5: **endwhile**

Experiments:

DC relaxation: negative semi-definite compatibility

We now introduce a new DC relaxation of our objective function that takes advantage of the structure of the problem. Specifically, the convex problem to solve at each iteration does not depend on the filter-based method computations. We rewrite the problem as

$$S(y) = \phi^T y - y^T(\mu \otimes I_{N)} y^T + y^T(\mu \otimes \sum_m K^{(m)} y. \quad (16)$$

LP Relaxation:

This section presents an accurate LP relaxation of the energy minimization problem and our method to optimize it efficiently using sub gradient descent.

Relaxation:

To simplify the description, we focus on the Potts model. However, our approach can easily be extend to more general pairwise potentials by approximating them using hierarchical Potts model. We define the following notation:

$K_{a,b} = \sum_m \omega^{(m)} k^{(m)}(f_a^{(m)}, f_b^{(m)}), \sum_a = \sum_{a=1}^{N} and \sum_{b<a} = \sum_{b=1}^{a-1}$. with these notations, a LP relaxation of equation(5) is

$\min S_{LP}(y) =$

$ai\phi aiyai + ab \neq aiKa,byai - yb(i)2,$

$s.t\ s \in \mu.$ (17)

The feasible set remains the same as the one we had for

Full computation **Half computations**

the QP and DC relaxations. In the case of integer solutions, $S_{LP}(y)$ has the same value as the objective function of the IP described in equation(5). The unary term is the same for both formulations. The pairwise term ensures that for every pair of random variables X_a,X_b, we add the cost $K_{a,b}$ associated with this edge only if they are not associated with the same labels.

Reformulation:

The absolute value in the pair wise term of equation(5) prevents us from using the filtering approach. To address this issue, we consider that for any given label i, the variables $y_a(i)$ can be sorted in a descending order: $a \geq b \Longrightarrow y_a(i) \leq y_b(i)$ this allows us to rewrite the pairwise term of the objective function(17) as:

$$\sum_a \sum_{b \neq a} \sum_i K_{a,b} \frac{|y_a(i) - y_b(i)|}{2} = \sum_i \sum_a \sum_{b>a} K_{a,b} y_{a(i)} - \sum_i \sum_a \sum_{b<a} K_{a,b} y_{b(i)} \quad (18)$$

Sub gradient

From Equ.(18) we rewrite the subgradient:

$$\frac{\partial S_{LP}}{\partial y_c(k)} = \phi_c(k) + \sum_{a>c} K_{a,c} - \sum_{a<c} K_{a,c} \quad (19)$$

Note that in this expression, the dependency on the variable **y** is hidden in the bounds of the sum because we assumed that $y_a(k) \leq y_c(k) for\ all\ a > c$. For a different value of **y,** the elements of **y** would induce a different ordering and the terms involved in each summation would not be the same.

Sub gradient Computation:

What prevents us from evaluating (20) efficiently are the two sums, one over an upper triangular matrix ($\sum_{a>c} K_{a,c}$) and one over a lower triangular matrix ($\sum_{a<c} K_{a,c}$). As opposed equation(6), which computes terms $\sum_{a,b} K_{a,b} v_b$ for all a using the filter-based method, the summation bounds here depend on the random variable we are computing the partial derivative for. While it would seems that the added sparsity provided by the upper and lower triangular matrices would simplify the operation, it is this sparsity itself that prevents us from interpreting the summations as convolution operations. We alleviate this difficulty by designing a novel divide-and-conquer algorithm. We describe our algorithm for the case of the upper triangular matrix. However, it can easily be adapted to compute the summation corresponding to the lower triangular matrix. We present the intuition behind the algorithm using an example.

If we consider an example. A rigorous development can be found $a, c \in \{1,2,3,4,5,6\}$ and the terms we need to compute for a given label are:

$$\begin{bmatrix} \sum_{a>1} K_{a,1} \\ \sum_{a>1} K_{a,2} \\ \sum_{a>1} K_{a,3} \\ \sum_{a>1} K_{a,4} \\ \sum_{a>1} K_{a,5} \\ \sum_{a>1} K_{a,6} \end{bmatrix} = \begin{bmatrix} 0 & K_{2,1} & K_{3,1} & K_{4,1} & K_{5,1} & K_{6,1} \\ 0 & 0 & K_{3,2} & K_{4,2} & K_{5,2} & K_{6,2} \\ 0 & 0 & 0 & K_{4,3} & K_{5,3} & K_{6,2} \\ 0 & 0 & 0 & 0 & K_{5,4} & K_{6,4} \\ 0 & 0 & 0 & 0 & 0 & K_{6,5} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

We propose a divide and conquer approach that solves this problem by splitting the upper triangular matrix U. The top-left and bottom-right parts are upper triangular matrices with half the size. We solve these sub problems recursively. The total complexity to compute this sum is $O(N \log(N))$. The complexity associated with taking a

gradient step is $O(MN\log(N))$.

The algorithm that we introduced converges to a global minimum of the LP relaxation. By using the rounding procedure introduced by Kleinberg and Tardos,it has a multiplicative bound of 2 for the dense CRF labeling problem on Potts models and O(log(M)) for metric pairwise potentials.

A Filter-based method Approximation:

We observe that for the variances considered in this paper and without using the normalization by Krahenbuhl and Koltun, the results given by the permutohedral lattice is a constant factor away from the value computed by brute force in most cases. As can be seen in Figure in the case where we Compute $\sum_{a,b} K_{a,b}1$. the left graph, the ratio between the value obtained by brute force and the value obtained using the permutohedral lattice is 0:6 for large enough images. On the other hand, for a different value of the input points where we compute $\sum_{b>a} K_{a,b} - \sum_{b<a} K_{a,b}$, the right graph, we get a ratio of 0:48 between the two results. The case where we consider a variance of 50 is special. We know that the highest the variance value, the worst the approximation of the permutohedral is. If the experience on the full computation is conducted on an image of size 320 213, the ratio between the brute force approach and the permutohedral lattice is 0:633. At the same time is also worth noting that in all these results, if we consider the outputs as vectors, as is done when computing our gradients, the vectors given by the brute force and the ones given by the permutohedral lattice are collinear for all image size and all variances. We can thus expect that for other input values, the direction of gradient provided by the permutohedral lattice is correct, but the norm of this vector may be incorrect.

Compute $\sum_{a,b} K_{a,b}1$. the left graph, the ratio between the value obtained by brute force and the value obtained using the permutohedral lattice is 0:6 for large enough images. On the other hand, for a different value of the input points where we compute $\sum_{b>a} K_{a,b} - \sum_{b<a} K_{a,b}$, the right graph, we get a ratio of 0:48 between the two results. The case where we consider a variance of 50 is special. We know that the highest the variance value, the worst the approximation of the permutohedral is. If the experience on the full computation is conducted on an image of size 320 213, the ratio between the brute force approach and the permutohedral lattice is 0:633. At the same time is also worth noting that in all these results, if we consider the outputs as vectors, as is done when computing our gradients, the vectors given by the brute force and the ones given by the permutohedral lattice are collinear for all image size and all variances. We can thus expect that for other input values, the direction of gradient provided by the permutohedral lattice is correct, but the norm of this vector may be incorrect.

The optimal step size can be computed and that this does not introduce any additional call to the filter-based method.

The problem to solve is

$$\underset{\alpha \in [0,1]}{argmin} S_{cvx}(y + \alpha(s-y)) \quad (21)$$

The definition of S_{cvx} is

$$S_{cvx}(y) = (\phi - d)^T y + y^T(\Psi + D)y \quad (22)$$

Solving for optimal value of α amounts to solving a second order polynomial:

$$\begin{aligned} S_{cvx}(y + \alpha(s-y) &= (\phi - d)^T(y + \alpha(s-y) \\ &+ (y + \alpha(s-y)^T(\Psi + D)y + \alpha(s-y)), \\ \alpha^2[(s-y)^T(\Psi + D)(s-y)] &+ \alpha[(\phi - d)^T(s-y) + \\ 2y^T(\Psi + D)(s-y)] &+ (\phi - d)^T y + y^T(\Psi + D)y, \quad (23) \end{aligned}$$

Whose optimal value is given by

$$\alpha^* = -\frac{1}{2}\frac{[(\phi - d)^T(s-y) + 2y^T(\Psi + D)(s-y)]}{(s-y)^T(\Psi + D)(s-y)} \quad (24)$$

The dot products are going to be linear in complexity and efficient. Using the filtering approach, the matrix-vector operation are also linear in complexity. In terms of run-time, they represent the costliest step so minimizing the number of times that we are going to perform them will gives us the best performance for

Our algorithm. We remind the reader of the expression of the gradient used at an iteration is:

$$\nabla S_{cvx}(y) = (\phi - d) + (\Psi + D)y \quad (25)$$

So by keeping intermediary results of the gradient's computation we don't need compute $(\Psi + D)y$, having already performed this operation once. The other matrix-vector product that is necessary for obtaining the optimal step-size is $(\Psi + D)s$. However, this computation can be reused. The updated rules that we follow are:

$$y^{t+1} = y^t + \propto (s - y^t) \quad (26)$$

At the following iteration, to obtain the gradient, we will need to compute:

$$\begin{aligned} (\Psi + D)y^{T+1} &= (\Psi + D)(y^t + \propto (s - y^t), \\ &= (1 - \propto)(\Psi + D)y^t + \propto (\Psi + D)s. \quad (27) \end{aligned}$$

All the matrix-vector product of this equation have already been computed. This means that no call to the

filter-based method will be required.

Convex problem in the restricted DC relaxation:

Two difference-of-convex decompositions of the objective function are presented in the paper. On the other hand, in the case of negative semi-definite compatibility functions, a decomposition suited to the structure of the problem is available. Using this decomposition, the convex problem to solve CCCP will be the following:

$$\min(\phi^T + g^T)y - y^T(\mu \otimes I_N)y,$$
$$s.t\ y \in \mu \quad (28)$$

The problem to solve for each pixel are, using the a subscript to refer to the subset of the vector elements that corresponds to the random variable a:

$$\min(\phi_a^T + g_a^T)\, y_a - y_a^T \mu y_a,$$
$$s.t \quad y_a \geq 0$$
$$y_a^T \mathbb{1} = 1. \quad (29)$$

These problems can also be solved using the Frank-Wolfe algorithm, with efficient conditional gradient computation and optimal step size. The only difference is that in that case, no filter-based method will need to be used for computation. CCCP on this DC relaxation will therefore be much faster than on the generic case, an improvement gained at the cost of generality.We also remark that the guarantees of CCCP to provide better results at each iteration does not require to solve the convex problem exactly. It is sufficient to obtain a value of the convex problem lower that the initial estimate. Therefore, the inference may eventually be sped-up by solving the convex problem approximately instead of reaching the optimal solution.

LP objective reformulation

This section presents the reformulation of the pairwise part of the LP objective. We first introduce the following equality:

$$\sum_a \sum_{b>a} K_{a,b} y_b(i) = \sum_a \sum_{b<a} K_{a,b} y_b(i) \quad (30)$$

It comes from the symmetry of K.

Using the above formula, considering the recording has already been done, we can write the pairwise term of (18) as:

$$\sum_a \sum_{b\neq a} \sum_i K_{a,b} \frac{|y_a(i)-y_b(j)|}{2},$$
$$= \sum_i \sum_a \sum_{b>a} K_{a,b} \frac{|y_a(i)-y_b(i)|}{2} - \sum_i \sum_a \sum_{b<a} K_{a,b} \frac{|y_a(i)-y_b(i)|}{2}, \quad (31)$$
$$= \sum_i \sum_a \sum_{b>a} K_{a,b} y_a(i) - \sum_i \sum_a \sum_{b<a} K_{a,b} y_a(i).$$

It is important to note in these equations, the ordering between a and b used in the summation is dependent on the considered label i.

LP Divide and Conquer:

We are going to present an algorithm to efficiently compute the following:

$$\forall k \sum_{j>k} K_{k,j}, \quad (32)$$

For j and k being 1 and N. for the sake of simplicity, we are going to consider N as being even. The odd case is very similar. Considering h-N/2, we can rewrite the original sum as:

$$\sum_{j>k} K_{k,j} = \begin{cases} \sum_{j>k} K_{k,j} & if\ k > h \\ \sum_{j>k} K_{k,j} & if\ k \leq h \end{cases}$$
$$= \begin{cases} \sum_{j>k} K_{k,j} & if\ k > h \\ \sum_{j\leq h} K_{k,j} + \sum_{j>h} K_{k,j} & if\ k \leq h \end{cases} \quad (33)$$

The two elements can be obtained by recursion using sub-matrices of K which have half the size of the current size of the problem. So we have a recursive algorithm that will have a depth of $\log(N)$ and for which all level takes $O(N)$ to compute. We can use it to compute the requested sum $\forall k$ in $O(N \log(N))$.

V.Results

The results of the proposed approach along with other recent methods are shown in Table. The results shown do not include any post-processing methods. The results illustrate that our approach achieves state-of-the-art performance in crowd count.

Method	*Mean Absolute Error*
Learning to Count	*493.4*
Density-aware Detection	*655.7*
FHSc	*468.0*
Cross-Scene Counting	*467.0*
Proposed	**452.5**

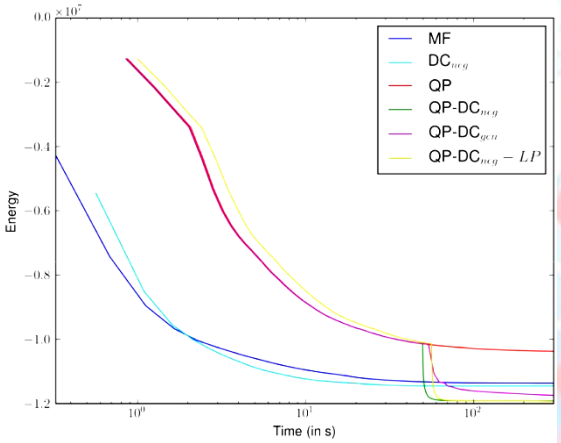
We also show the predicted count for each image in the dataset along with its actual count in Fig4. For most of the images, the predicted count lies close to the actual count. However, we observe that the proposed approach tends to underestimate the count in cases of images with more than 2500 people. This estimation error could possibly be a con-sequence of the insufficient number of training images with such large crowds in the dataset.

We now demonstrate the benefits of using continuous relaxations of the energy minimisation problem on two

applications: stereo matching and semantic segmentation. We provide results for the following methods: the Convex QP relaxation (QP_{cvx}), the generic and negative semi-definite specific DC relaxations (DC_{gen} and DC_{neg}) and the LP relaxation (LP). We compare solutions obtained by our methods with the mean- field baseline (MF).

Stereo matching

We compare these methods on images extracted from the Middlebury stereo matching dataset. The unary terms are obtained using the absolute difference matching function ofThe pixel compatibility function is similar to the one used by Kr•ahenb•uhl and Koltun and is described in Appendix G. The label compatibility function is a Potts model.



(a) Runtime comparisons b) Final Energy achieved

In the case of negative semi-definite potentials, the specific DC_{neg} method is as fast as mean- field, while additionally providing guarantees of monotonous decrease.(Best viewed in colour)

We observe that continuous relaxations obtain better energies than their mean-field counterparts. For a very limited time-budget, MF is the fastest method, although DC_{neg} is competitive and reach lower energies. When using LP, optimising a better objective function allows us to escape the local minima to which DC_{neg} converges. However, due to the higher complexity and the fact that we need to perform divide-and-conquer separately for all labels, the method is slower. This is particularly visible for problems with a high number of labels. This indicates that the LP relaxation might be better suited to ne-tune accurate solutions obtained by faster alternatives. For example, this can be achieved by restricting the LP to optimise over a subset of relevant labels, that is, labels thatare present in the solutions provided by other methods. Qualitative results for the Teddy image can be found in Figure 2 and additional outputs are present in Appendix H. We can see that lower energy translates to better visual results: note the removal of the artifacts in otherwise smooth regions (for example, in the middle of the sloped surface on the left of the image).

Method	Final energy
MF	***-1.137e+07***
DCneg	***-1.145e+07***
QP	***-1.037e+07***
QP-DCneg	***-1.191e+07***
QP-DCgen	***-1.175e+07***
QP-DCneg***-LP***	-1.193e+07

Image Segmentation

We now consider an image segmentation task evaluated on the PASCAL VOC 2010 dataset. For the sake of comparison, we use the same data splits and unary potentials as the one used by Kr•ahenb•uhl and Koltun. We perform cross-validation to select the best parameters of the pixel compatibility function for each method using Spearmint.

The energy results obtained using the parameters cross validated for DC_{neg} are given in Table 1. MF5 corresponds to mean- field ran for 5 iterations as it is often the case in practice.

Once again, we observe that continuous relaxations provide lower energies than mean field based approaches. To add significance to this result, we also compare energies image-wise. In all but a few cases, the energies obtained by the continuous relaxations are better or equal to the mean- field ones. This provides conclusive evidence for our central hypothesis that continuous relaxations are better suited to the problem of energy minimization in dense CRFs.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Ravikumar, P., La erty, J.: Quadratic programming relaxations for metric labeling and Markov random field MAP estimation. In: ICML. (2006)

[2] Kleinberg, J., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. JACM (2002)
[3] Kumar, P., Kolmogorov, V., Torr, P.: An analysis of convex relaxations for MAP estimation. In: NIPS. (2008)
[4] Chekuri, C., Khanna, S., Naor, J., Zosin, L.: Approximation algorithms for the metric labeling problem via a new linear programming formulation. In: SODA. (2001)
[5] Kr•ahenb•uhl, P., Koltun, V.: Efficient inference in fully connected CRFs with gaussian edge potentials. In: NIPS. (2011)
[6] Tappen, M., Liu, C., Adelson, E., Freeman, W.: Learning gaussian conditional random fields for low-level vision. In: CVPR. (2007)
[7] Koller, D., Friedman, N.: Probabilistic graphical models: principles and techniques. (2009)
[8] Adams, A., Baek, J., Abraham, M.: Fast high-dimensional ltering using the permutohedral lattice. Eurographics (2010)
[9] Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval research logistics quarterly (1956)
[10] Kr•ahenb•uhl, P., Koltun, V.: Parameter learning and convergent inference for dense random fields. In: ICML. (2013)
[11] Baque, P., Bagautdinov, T., Fleuret, F., Fua, P.: Principled parallel mean- field inference for discrete random fields. In: CVPR. (2016)
[12] Vineet, V., Warrell, J., Torr, P.: Filter-based mean- field inference for random fields with higher-order terms and product label-spaces. IJCV (2014)
[13] Ladicky, L., Russell, C., Kohli, P., Torr, P.: Graph cut based inference with co-occurrence statistics. In: ECCV. (2010)
[14] Kohli, P., Kumar, P., Torr, P.: P3 & beyond: Solving energies with higher order cliques. In: CVPR. (2007)
[15] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
[16] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR. (2015)
[17] Schwing, A., Urtasun, R.: Fully connected deep structured networks. CoRR (2015)
[18] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: ICCV. (2015)
[19] Zhang, Y., Chen, T.: Efficient inference for fully-connected CRFs with stationarity. In: CVPR. (2012)
[20] Wang, P., Shen, C., van den Hengel, A.: Efficient SDP inference for fully-connected CRFs based on low-rank decomposition. In: CVPR. (2015)
[21] Goemans, M., Williamson, D.: Improved approximation algorithms for maximum cut and satis ability problems using semi definite programming. JACM (1995)
[22] Lacoste-Julien, S., Jaggi, M., Schmidt, M., Pletscher, P.: Block-coordinate frank-wolfe optimization for structural svms. In: ICML. (2013)
[23] Yuille, A., Rangarajan, A.: The concave-convex procedure (CCCP). NIPS (2002)
[24] Sriperumbudur, B., Lanckriet, G.: On the convergence of the concave-convex procedure. In: NIPS. (2009)
[25] Kumar, P., Koller, D.: MAP estimation of semi-metric MRFs via hierarchical graph cuts. In: UAI. (2009)
[26] Condat, L.: Fast projection onto the simplex and the l1 ball. Mathematical Programming (2015)
[27] Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV (2002)
[28] Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PAS-CAL Visual Object Classes Challenge 2010 (VOC2010) Results
[29] Snoek, J., Larochelle, H., Adams, R.: Practical bayesian optimization of machine learning algorithms. In: NIPS. (2012)
[30] Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI (2001)
[31] Bartal, Y.: On approximating arbitrary metrices by tree metrics. In: STOC. (1998)
[32] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In IEEEConference on Computer Vision and Pattern Recognition, 2008.
[33] G. J. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In IEEEConference on Computer Vision and Pattern Recognition, 2006.
[34] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. arXivpreprint arXiv:1412.7062, 2014.
[35] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In IEEE ComputerSociety Conference on Computer Vision and Pattern Recognition, 2005., volume 1, pages 886{893. IEEE,2005.
[36] J. Ferryman and A. Ellis. Pets2010: Dataset and challenge. In IEEE International Conference onAdvanced Video and Signal Based Surveillance (AVSS), 2010.
[37] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. arxiv:1502.02734, 2015.
[38] J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Ca e: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014.
[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.