Image denoising using CRF

Aditya Dhall
2581439
s8addhal@stud.uni-saarland.de

Pramod Rao
2581268
s8prraoo@stud.uni-saarland.de

Kaustubh Sakhalkar
2581456
s8kasakh@stud.uni-saarland.de

Abstract

Probabilistic models like CRFs and MRFs have been employed for low level vision tasks like denoising, optical flow estimation and corner detection. Recently deep learning methods have made several advancements even in low level vision tasks. DNNs still find it difficult to fine tune certain pixels at edges in segmentation or denoising. This leads to small blobs at edges or pixelated artefacts in denoising problems. In this project we attempt to solve this problem by taking best of both worlds of probabilistic methods and Deep Neural Networks (DNNs). To this end we use a FCN as backbone for denoising and CRFasRNN [1] for refinement of the images. We show how CRFasRNN can be adapted to regression task like denoising and present our results in the end.

1 Introduction

Noise is ubiquitous in all image acquisition processes and a lot of effort has been invested in designing denoising methods. The interest in denoising methods has increased as we characterize new sources of noise. Understanding noise in images not only helps us to remove artefacts but also detect forgeries in images by studying the noise fingerprint of a camera or inspecting the fourier transform of the image signal. Noise in images can arise from either hardware when we convert sensor data to digital image or physical properties of light. A common source of noise is the quantization of signal when we convert from sensor data to image data or the lens shutter affecting the time at which the sensor records the signal. Heating of the sensor creates dark shot noise which is related to thermal noise in the camera. Lens shutter causes irregularity in the time at which photons arrive on the sensor. Another source of noise is the fluctuations in surrounding temperature which causes changes in the refractive index of air thereby fluctuating the direction of light through thin air and creating color artefacts. Distorted lenses lead to chromatic aberrations. Image compressions or interpolations can enhance the noise in the images.

The most realistic noise model for images is Poisson with gaussian i.e. a combination of additive and multiplicative noise but to keep things simple we only use additive noise models of salt & pepper noise with gaussian noise of mean 0 and known standard distribution. CRF based techniques can be extended to any noise model as long as the logarithm of the distribution is differentiable. In practice the amount of noise present is not known. The classical methods for denoising are broadly categorrized into two parts viz. with apriori knowledge (CRF) and without apriori knowledge (PDE based). Gaussian filters basically exploit the local smoothness assumption of the image. Other PDE based techniques can be thought as a diffusion process. Global smoothness assumptions often lead to loss of edges so non-linear diffusion techniques were proposed. Other types of method include variational minimizers which rely on energy minimization.

course project saarland. Under review.

Probabilistic methods like MRFs and CRFs are well known. Denoising is also done in transformed domains like wavelets and fourier. In fourier domain the frequencies above a certain range are clamped. In wavelets, the image gets transformed into several wavelet components. The wavelet coefficients are modified, typically shrinked based on a function. The image is the reconstructed from the modified wavelets. But these methods work in another domain which may not preserve the natural image statistics. Probabilistic methods give the best results among classical techniques.

More recently variational autoencoding methods have been introduced for denoising. Deep learning methods do not set a prior but directly learn the denoising function from noisy and denoised image pairs. A simple 5 layer network can give comparable results to classical algorithms like BM3D. As a classical denoiser BM3D [2] is very popular because it gives good results with less computational overhead and is based on patchwise statistics. There have been several attempts to denoise images with deep neural networks. Stacking auto encoders is also a well known method for denoising [3]. For a detailed literature review of deep learning denoising methods we refer the reader to [4].

In our project we try to use the best of both worlds of probabilistic denoising methods with a priori knowledge and deep neural networks. Then use the CRFasRNN model [1] which was originally proposed for segmentation to refine the result of the deep learning model. We assume an additive noise model $y = I + n$ where y is the noisy image and we try to learn clean images I from y. In the following section 2 we detail about CRFasRNN.

2 Background

2.1 CRFasRNN

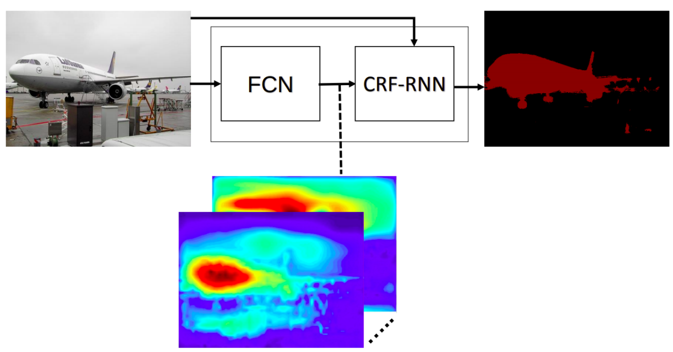

Figure 1: CRFasRNN [1] pipeline for segmentation

Pixel level vision problems or low level vision problems are an active area of research in image processing. Large deep learning models find it difficult to refine small patch wise changes so CRFasRNN [1] was proposed by Zheng et al for refining segmentation which is plugged in as a part of CNN. The primary motivation of CRFasRNN was to tackle the small clusters of misclassified pixels that form at the class boundaries. We find similar problems with denoising methods where using deep neural networks alone causes distortion in high frequency regions of the image. The general purpose of a probabilistic random field is to refine the output at a pixel. A fully connected CRF is used that tries to minimize the a Gibbs energy of the form:

$$P(X = x|I) = \frac{1}{Z(I)} \exp(-E(x|I))$$

Where I is the input image and Z is the partition function. They use a unary potential to measure the similarity between the original classification and the neighbouring classification. The key idea in CRFasRNN is to use the CRF as a stack of CNN.

Algorithm 1 Mean-field in dense CRFs broken down to common CNN operations.

$Q_i(l) \leftarrow \frac{1}{Z_i}\exp\left(U_i(l)\right)$ for all i ▷ Initialization

while not converged **do**

$\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m ▷ Message Passing

$\check{Q}_i(l) \leftarrow \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$ ▷ Weighting Filter Outputs

$\hat{Q}_i(l) \leftarrow \sum_{l' \in \mathcal{L}} \mu(l, l') \check{Q}_i(l')$ ▷ Compatibility Transform

$\breve{Q}_i(l) \leftarrow U_i(l) - \hat{Q}_i(l)$ ▷ Adding Unary Potentials

$Q_i \leftarrow \frac{1}{Z_i}\exp\left(\breve{Q}_i(l)\right)$ ▷ Normalizing

end while

Figure 2: One iteration of CRF as RNN

The unary potential U_i is assumed to come from the FCN then a softmax is applied on the unary potentials to get the CRF distribution. For message passing two gaussian kernels are used. One as a spatial kernel and another as bilateral kernel. Then a weighted sum of the filtered outputs is done which can be assumed as a 1×1 convolution. They have used individual weights for each class label which we change for denoising. A compatibility transform is applied on the unary potentials. This process is iterated multiple times and the parameters are shared hence the name recurrent CRF. Figure 2 describes this formally.

3 Approach

To perform image denoising we have opted for a FCN3 network and CRF implementation of [1] to improvise on the result from FCN3 denoiser. Our FCN3 is a sequential model of three convolutional block comprising of convolution layer, batch normalization and ReLU activation. We preferred not to use the FCN8 model as suggested in [1] as it would be difficult to train small subset of low resolution images due to its high model capacity.

As discussed in the previous section [1] comprises of two subsequent models first the FCN8 to predict the semantic segmentation mask and a unrolled RNN model to perform mean field inference on the output from FCN8 to improve the results. Unlike them we don't perform end to end training, we prefer to train the FCN3 first and then train the CRF module separately.

[1] is used for the purpose of semantic segmentation but our task is image denoising, so we need to redefine our problem accordingly to make it compatible. So we will be formulating the task of image denoising that is a regression task as classification task. As a pixel can attain only 256 distinct values in the uint8 format, we will consider classification task for each pixel with 256 classes. Similar to the semantic segmentation mask which captures the class to which a particular pixel belongs to, we will transform the output from FCN3 network and transform it to arrays of logits for each pixel. Now as we have formulated the input to the CRF module inline with that of a typical segmentation task, we can proceed as proposed by [1].

Now we have a denoised image from the FCN3 network and a logits matrix based on the denoised image, both will be input to the CRF module. CRF module uses the denoised image given as the input to initialize the coefficients of the spatial and bilateral filters using the spatial location and pixel value as the feature set for each pixel. As the CRF is fully connected receptive field of each filter spans the whole image, it becomes infeasible to use brute-force implementation of these filters. Instead Permutohedral lattice implementation is used which is capable of computing the filter response in O(N), where N is the number of pixels. Only set of learnable parameters are class wise weights of the

	FCN3	FCN3 + CRF
PSNR	6.21	6.79

Table 1: Mean PSNR result for FCN3 and FCN3 + CRF model over the dataset

filters used and the compatibility matrix defined by [1]. Compatibility matrix penalizes the allocation of different labels to pixel with similar properties.

With each iteration logits are first normalized then the filters are applied and their weighted sum is computed. Compatibility matrix is multiplied with the filter output. This message passing output is subtracted from the unary potential ,i.e., logits and such iterations are repeated. We opt for 5 iterations to avoid the case of vanishing & exploding gradient. Output of this CRF module is similar to logits matrix given as the input, now to train the parameters cross entropy loss for each pixel is used.

As proposed in our project proposal we analyze the effect of denoising by MRF technique we built in our assignment-03. The output from the FNC3 network is used as an initialisation point for the MRF-based denoising method for Gaussian and student-t distribution priors.

4 Results

Our proposed method is successful on image denoising task. We evaluate our approach using 4500 random grayscale images from PASCAL VOC 2012 dataset [5] by adding salt-and-pepper noise which could be used to model the defect in the CCD or transmission of images. The main motivation to use the dataset was to find a balance between computation costs and sufficiently high resolution to train our FCN network. During the course of our experiments we use images of 64x64 resolution, further, we observed that low-resolution images such as MNIST [6] and CIFAR-10 [7] were difficult to train.

As mentioned before we trained the FCN3 model first to denoise the noisy images. Then train the CRF model to enhance the performance of results given by FCN3 model. Peak to signal ratio (PSNR) will be used as the evaluation metric to evaluate the performance of different models. Figure 3 shows the qualitative result left image depicts the noisy image, centre image is ouput FCN3 model and the right one is given by CRF model. Visually there isn't much difference between the output of FCN3 and CRF model as the increment in PSNR value is not significant shown in Table 1. FCN3 + CRF training settings are: FCN3 trained for 15 epochs with learning rate $1e-4$ and MSE loss function whereas CRF model is trained for 5 epochs with learning rate $1e-6$ and Cross entropy loss function. Convergence in the case of CRF model is quite slow, [1] suggest to use 1e-13 as learning rate. Certain anomalies can also be observed in the results from CRF model as shown in Figure 5 which will be discussed in the next section.

(a) Noisy Image (b) Result with FCN3 (c) Result with FCN3 + CRF

Figure 3: Qualitative result of image denoising

Our FCN3 network provides a good initialisation point for MRF-based denoising. We evaluate on few randomly chosen images and observe that on using MRF-based denoising method on FCN3 output consistently improves the images. One such example is show in the Figure 4 with Gaussian and student-t priors.

Figure 4: MRF Results. Starting from left, first image is the noisy input, second image is the output from FCN3 (PSNR=16.6075), followed by the MRF output with Gaussian prior (PSNR = 17.1962) and right most is the MRF denoised output with student-t prior (PSNR = 17.1919). Although the PSNR value for student-t prior is slightly lower than the former, we can observe that the image is much sharper.

5 Discussion and Conclusion

We observed that there is no significant improvement in the PSNR value with introduction of CRF model but our results were comparable with that of MRF. As the learning rate suggested by [1] is 1e-13 but when we tried to use this the model convergence was too slow and due to computational constraints we couldn't use this to train our model. So we decided to use 1e-6 as the learning rate but this actually introduced artefacts in some images shown in Figure 5. These artefacts are introduced due to the vanishing or exploding gradients of the typical RNN cell operations for performing the mean field inference iterations.

Use of much lower learning rate and number of inference iterations could remove these artefacts. Also rigorous training could make the model converge with lower learning rate. Also replacement of rnn-cell like operations with LSTM-cell like operation could help in overcoming these drawbacks. We restricted ourselves to use 64×64 images as the permutohedral implementation of filters in [1] appeared to be the bottleneck and it was computational infeasible to train with 256×256 given our limited resources.

We articulated the denosing problem as pixel-wise classification problem by modelling the CRF as a recurrent neural network. We proposed FCN3 network as first stage denoising module and provide a good initial point for further processing. Finally, we show consistent improvements in both MRF and CRF-based denoising methods for PASCAL VOC 2012 images.

Figure 5: Artefacts introduced by the CRF model

References

[1] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[2] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715–1728, 2011.

[3] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[4] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 2020.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[6] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[7] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.