# Application of machine learning methods in tomographic reconstruction with low radiation dosage

**CS 754: Project Report**

by

**Yadnyesh Patil (193050067)**

under the guidance of

**Prof. Ajit Rajwade**

M.Tech. I, CSE

Indian Institute of Technology, Bombay

# Contents

# Chapter 1

# Introduction and approach

Low radiation tomography holds very important practical significance in medical applications. But, acquisition noise level increases greatly and reconstructions in this regime becomes challenging. This is because, imaging model in terms of intensity of X-rays is given as:

$$y \sim Poisson(I_0\, e^{-\mathbf{\Phi} x}) + \eta \tag{1.1}$$

Thus, in lowdose regime, CT model is associated with Poisson-Gaussian noise. Hence measurements $y$, are expressed as summation of 2 random variables, Poisson random variable with mean $I_0\, e^{-\mathbf{\Phi} x}$) and Gaussian random variable, $\eta$ with fixed and signal independent standard deviation $\sigma$. $\mathbf{\Phi}$ is the forward model for tomographic projections and $x$ is the image or the test object. Signal-to-Noise ratio of poisson variable is $\sqrt{I_0\, e^{-\mathbf{\Phi} x}}$. In lowdose, $I_0$ is less, therefore low dose reconstructions are associated with large noise.

Since measurements are noisy in lowdose regime, same technique of calculating weights map directly in spatial domain can't be done unlike fewviews case. Therefore first, the bins of test measurements that are associated with the change are detected by using same angle specific eigenspace of the templates' noisy measurements. Probability of each of the bin values being a part of new regions is calculated by hypothesis testing. These probabilities associated with each bin value and an angle are then backprojected to get the weights map. This weights map is used in reconstruction using prior templates.

## 1.1 Algorithm for computation of weights map by existing method

Following algorithm is adopted from[2]:

1. Let $x_1, x_2 ... x_N$ be $N$ high quality templates.
2. Simulate noiseless measurements using low dose intensity as: $y_i = I_0 e^{-\mathbf{\Phi} x_i}$ corresponding to each $x_i$, where i = 1...N

3. Let $y_i^j$ represent tomographic projection of $i^{th}$ template in $j^{th}$ angle, where j = 1...A. Let $E_j$ represent the eigenspace formed from measurements of all templates in $j^{th}$ angle. So, we have, $E_1, E_2...E_A$ measurement eigenspaces.

4. $y_{test}^j$ represents the noisy measument of test image in $j^{th}$ angle. Project $y_{test}^j$ on $E_j$ and calculate eigen-coefficents $\alpha_j$ of eigenvector matrix $V_j$ as :

$$\alpha_j = (V_j)^T(y_{test}^j - \mu_j)$$

5. Compute projections of the noisy measurements into the measurement eigenspaces as follows:

$$\hat{y}_{test}^j = \mu_j + V_j\alpha_j$$

6. Perform hypothesis testing on the quantity: $\sqrt{y_{test} + \frac{3}{8} + \sigma^2} - \sqrt{\hat{y}_{test} + \frac{3}{8} + \sigma^2}$ to detect the bins corresponding to new change. This is decided by the Z-test which gives probability (p-value) that the given sample is drawn from $\mathcal{N}(0, \frac{1}{4})$. The confidence level set is 95%.This idea is based on generalized Anscombe transform which states that, a random variable $s \sim Poisson(\lambda) + \eta$ with $\eta \sim \mathcal{N}(0, \sigma^2)$, then $\sqrt{s + \frac{3}{8} + \sigma^2}$ is approximately distributed as $N(\sqrt{\lambda + \frac{3}{8} + \sigma^2}, \frac{1}{4})$.

7. Take filtered back-projection of the vectors (corresponding to the measurement bins) with their p-values to get $W_{inlier}$ containing the locations of new changes in image domain.

8. Calculate $W$ matrix as:

$$W = \frac{1}{1 + W_{inlier}^2}$$

   this is done to allocate lower weight to new region and higher weight to older regions. This is followed by linear rescaling between 0 to 1.

## 1.2 Algorithm to compute weights map by machine learning methods

Existing method is fairly complex in the sense that, it involves using hypothesis test on every bin value in the measurement and its corresponding projection in measurement eigenspace. So, the method is comparatively slow. This can be avoided if a machine learning model is trained to learn a binary weights map i.e. the regions of actual new changes, given the residual image i.e. the difference between pilot reconstruction and its projection on high quality eigenspace. This training is done in following way:

1. Suppose we have $N$ templates. Treat one of the templates as a test image.

2. Simulate the lowdose noisy measurements of the selected test image and reconstruct using them to get a pilot reconstruction, $Test_{pilot}$

3. Project $T_{pilot}$ onto high quality eigenspace of $N-1$ templates to get $Test_{pilot,proj}$

4. Calculate: *Residual Image* = $||Test_{pilot} - Test_{pilot,proj}||$

5. Generate **input** to the machine learning model as patches from this residual image.

6. Generate **output** to train the model, as a binary image indicating regions of new changes, based on domain knowledge. Binary value of the centre of the patch will act as a label to the corresponding patch in the input. Model learn to assign a binary label to the given patch vector during training. Patch vector includes the information of grayscale values of pixels surrounding the centre of the patch. Corresponding label categorizes the centre of the patch as 1 (region of change) or 0(region of no change).

7. Train the model with this **input** and **output**.

8. Once model is trained, it can be tested on a new test image by creating a residual image of the new test. Same eigenspace is used during calculation of residual image that was used for training. Information from the output of the model will be a binary weights map $W$.

9. Use $W$ to improve the pilot reconstruction by using irradiation along with weighted prior reconstruction technique using $N-1$ templates.

Once model is trained it can be implemented on the test. This method is comparatively faster than hypothesis-test method.

## 1.3 Problem statement

Hence, the goal of this project is to explore how a machine learning method can be applied to generate the weights map in lowdose regime to achieve comparable or better results than that of the article[2].

# Chapter 2

# SVM

SVM is a supervised machine learning model that is used for classification. In this case, SVM model is used to perform the task of binary classification. SVM achieves this purpose by finding the best hyperplane that separates the two classes by maximizing the margin. SVMs work well with any dimensional data points. Non linearly separable cases are handled by transforming the data points into higher dimension in which they become separable. This is done very efficiently by using Kernel Trick, without actually doing the transformation. The Kernel Trick allows efficient computation of dot product of the between the 2 data points in higher dimension without transformation. Here, in this case RBF (Radial Basis Function) kernel is used as it can work any dimensional datasets. Important parameters that are tuned in the training are $C$ and $\gamma$.

The parameter $C$ represents miss-classification cost[1]. Its value decides the allowable error of mis-classification. Higher value of $C$ implies, a stricter model that classify the data well at the cost of smaller margin. Low value of $C$, implies lenient model that classify the data with higher margin at the cost of miss-classification.

The parameter $\gamma$ decides the spread of the kernel[1] i.e. the decision region. Low value of $\gamma$ implies the decision region is broad and farther points are also considered while calculating decision boundary. High value of $\gamma$ implies that decision region is small and only nearby points are considered for deciding decision boundary.

Th parameters $C$ and $\gamma$ are tuned by 10-fold cross validation for different combination of values. The training dataset was created as mentioned above in the procedure. The model was trained with weights map that was computed by following 2 ways:

## 2.1 Results of Manual selection of ROIs

In this method, the rectangular regions of change are selected based on the domain knowledge. A binary image indicating the regions of change in white and old regions in black, is formed. Such a image contains the information about the labels of the corresponding pixels in the residual image. SVM model is then trained with set of patchvectors and their corresponding label.

(a) Template 1 (b) Template 2 (c) Template 3 (d) Template 4 (e) Test image

Figure 2.1: Details of templates.(a),(b),(c) are used for high quality eigenspace,(d) is a validation image used for training purpose (e) is the test image with ROI shown in red box.

(a) Validation Image (b) Pilot recon (c) Pilot Projection (d) Residual image (e) Selected weight map

Figure 2.2: Training input and output for SVM. (a) is used to train the SVM model. (b) is the pilot reconstruction of (a) using noisy lowdose measurements using RNLLS. (c) is the projection of the pilot reconstruction on high quality eigenspace formed by templates:1,2 and 3. (d) is the residual image of (a). (e) is the manually selected weights map or labels to train SVM.

These results are obtained by considering low dose X-ray intensity of $I_0 = 2000$ and 360 number of angles. 1% of the mean of the poisson corrupted measurements is considered as

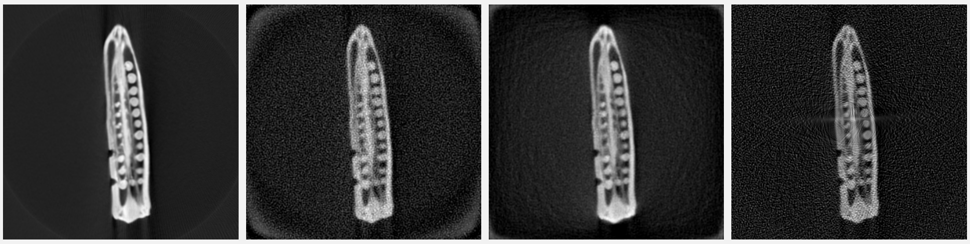
Gaussian noise. For weighted reconstruction using prior, $\lambda_1$ and $\lambda_2$ are chosen as 1 and 1200 respectively. Re-irradiation is carried out with standard X-ray dose intensity $I_0 = 5000$. Weiner filter with kernel 12 x 12 is used for adaptive filtering of noise.

(a) Unseen Test Image (b) Pilot recon (c) Pilot Projection (d) Residual image (e) Predicted weight map

Figure 2.3: Testing input and predicted output of SVM. (a) Test image. (b) is the pilot reconstruction of (a) using noisy lowdose measurements using RNLLS. (c) is the projection of the pilot reconstruction on high quality eigenspace formed by templates:1,2 and 3. (d) is the residual image of (a). (e) is the predicted weights map or labels by SVM model.

Comparing figure 2.3(d) and figure 2.3(e), predicted weights map captures the regions of changes reasonably well. However, it also neglects some changes in the top right portion of the middle section of okra. This is because, the model is trained by using manually generated weights map that doesn't cover all possible changes but only subset of them. This is avoided by auto-selection of the changes for generating weights map.

Comparing figure 2.3(b) and (c), there are two upper extra notches that are reflected in 2.3(d), that are major regions of changes. Figure 2.4 (c) and (d) reflects these changes clearly. Figure 2.4 (c) shows that using prior improves the overall image quality, but regions of changes are still not very sharp. Figure 2.4 (d) takes of this issue and shows enhanced quality of new regions of change.

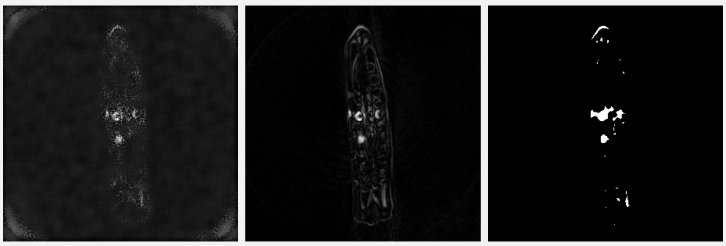


(a) Unseen Test Image (b) Pilot recon (c) Weighted prior recon (d) After re-irradiation

Figure 2.4: Reconstruction result with predicted weights.(c) is reconstruction using weights map and prior templates 1,2 and 3. (d) is the result obtained after re-irradiation in the regions in the weights map.

## 2.2 Results of Auto detection using thresholding

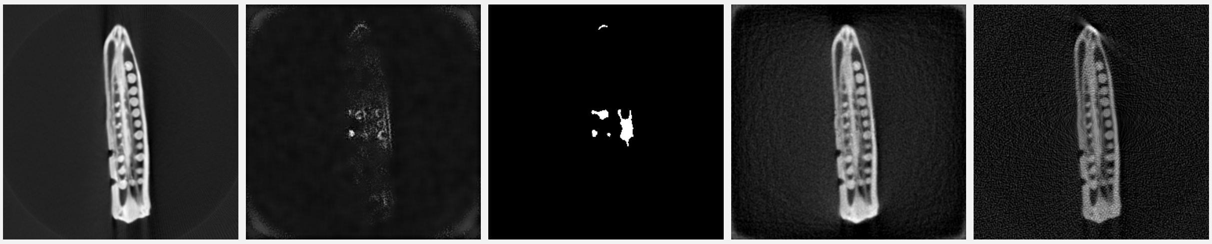
Main difference here is to train the model based on the weights map or labels that are not manually selected. This method is better since it is less probable to miss some of the changes. Actual inlier was generated by taking difference between a FBP pilot reconstruction of the noiseless measurements of template 4 and its projection on high quality eigenspace formed by templates 1,2 and 3. This inlier is then converted to binary map by choosing a global threshold. Therefore, it must be chosen carefully. In these experiments, global threshold is taken to be 0.3 in *imbinarize()* function.



(a) Residual image (b) Actual gray inlier (c) Binary inlier

Figure 2.5: SVM-auto selection. (a) is the residual image of template 4 i.e validation image used for training. (b) Actual gray inlier of the validation image. This is binarized with global threshold of 0.3 to get (c). (c) is used as weights or label map for training.

Comparing figure 2.6(b) and (c), nearly all regions have been detected. Figure 2.6(d) and



(a) Test image (b) Residual image (c) Predicted weights map (d) Weighted prior recon (e) After irradiation

Figure 2.6: SVM-auto selection. (a) is the test image. (b) is the residual image of the test image. (c) is the predicted weights map by the model. (d) is the result of reconstruction with using the weights map (c) and prior templates 1,2 and 3. (d) is the result after reirradiating the regions of changes predicted in (c).

(e) clearly highlight these changes.

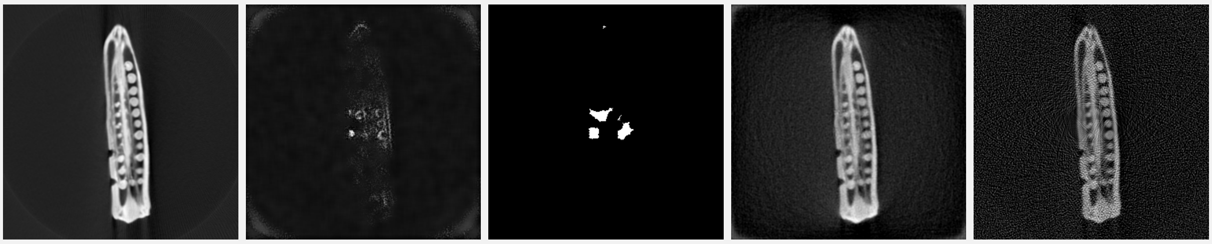
# Chapter 3

# CNN

Convolutional Neural Networks (CNNs) are a class of deep neural networks that are specifically effective in image classification problems. Structure of CNN is inspired by the way, images are recognized by human brain and visual cortex connected through complex network of neurons. Basic architecture of CNN involve input layer followed by series of hidden layers, followed by the output layer. Basic structure of CNN and parameters used are as follows[3]:

- *Input Layer*: Input layer is nothing but the vectorized image, in this case, 10x10 vectorized image.
- *Convolutional Layer*: Convolutional layer performs the convolution operation between the image pixels and the filters of user defined dimensions. Size and number of filters are hyper-parameters. Output of the convolution layer are called feature maps that contains the extracted features by the convolutional filters. These maps acts as the input to the pooling layer. In this case, 5 filters of size 3x3 were taken in each of the two convolutional layers.
- *Pooling layer*: Output of the convolutional layer is fed to the Pooling layer. Main task of pooling layer is to subsample the image by using functions like min(), max(), average(),etc. Output of the pooling layer can be again fed to convolutional layer to generate even more complex feature maps or it can be fed to the fully connected layer. Each of the 2 Max pooling layers used here, has the filter size of 2x2.
- *Fully Connected Layers*: These layers consist of series of traditional artificial neural network layers. Each neutron in a layer connects every other neuron in the next layer. Last layer of the fully connected network is the output layer that describes the output.

## 3.1 Results of Manual selection of ROIs



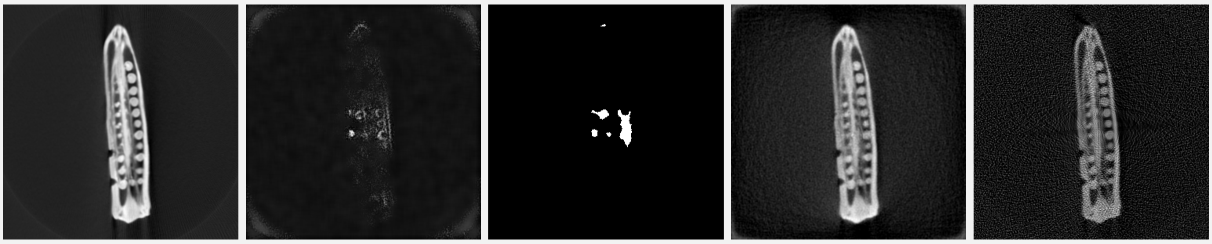
(a) Test image (b) Residual image (c) Predicted weights map (d) Weighted prior recon (e) After irradiation

Figure 3.1: CNN-manual selection of ROI. (a) is the test image. (b) is the residual image of the test image. (c) is the predicted weights map by the model. (d) is the result of reconstruction with using the weights map (c) and prior templates 1,2 and 3. (d) is the result after reirradiating the regions of changes predicted in (c).

In this method, CNN network is trained with input as patchvectors of the residual image of template 4 and output as the corresponding pixel values in the manually constructed weights map by selecting regions of changes based on domain knowledge.

Comparing figure 3.1(b) and (c), it can be observed that some regions in the middle section of the okra are not detected (just like in the above case of SVM). However, figure 3.1(d) shows that the extra notches present in the residual image have been rectified. Those regions are further improved in figure 3.1(e), by re-irradiating with standard dose of X-ray with intensity, $I_0 = 5000$ as opposed to low dose with intensity, $I_0 = 2000$.

## 3.2 Results of Auto detection using thresholding



(a) Test image (b) Residual image (c) Predicted weights map (d) Weighted prior recon (e) After irradiation

Figure 3.2: CNN-auto selection. (a) is the test image. (b) is the residual image of the test image. (c) is the predicted weights map by the model. (d) is the result of reconstruction with using the weights map (c) and prior templates 1,2 and 3. (d) is the result after re-irradiating the regions of changes predicted in (c).

In this method, CNN network is trained with input as patchvectors of the residual image of template 4 and output as the corresponding pixel values in automatically generated binary weights map. Same procedure is followed as mentioned above in 2.2.

Comparing figure 3.2(b) and (c), almost all the regions have been detected including some (like the top tip of the okra) that may not be of actual interest but are the potential regions of change due to significant intensity difference as denoted in residual image. However, figure 3.1(d) shows that the extra notches present in the residual image have been rectified. Those regions are further improved in figure 3.2(d), by re-irradiating with standard dose of X-ray with intensity, $I_0 = 5000$ as opposed to low dose with intensity, $I_0 = 2000$.

# Chapter 4

# Conclusion and analysis

Results show that the machine learning methods can be employed successfully to generate the weights map in the case of low dosage of X-rays.

**Good aspects of this method**:

- This method does not make any assumption of noise model. This method is unaware of the noise model assumed for simulating measurements.
- This method does not involve hypothesis testing and back projection of the p-values obtained from hypothesis testing. So, it is more robust.
- Existing method to generate weights map often suffer the problem of streak artefacts when p-values are back-projected. This method does not result in streak artefacts.
- This method is unlikely to result in any false negatives if trained with labels map obtained with low threshold.

**Bad aspects of this method**:

- Main challenge of this method is to create a training data. The training data should capture all the changes based on intensity difference that results in detecting all true changes in the test image but at the same time minimizing false positives. False positives are not a problem in terms of missing any change that my be very critical in medical perspective. However, large false positive increases radiation exposure if re-irradiation is employed for further clarity.
- While generating labels map, manual thresholding is done to decide the lables in *imbinarize()* function. Even if this value result in good weights (or labels) map, it doesn't guarantee absence of false negatives.

**Future work**:

This approach can be modified to use machine learning regression methods to get gray weights map. Gray weights map will eliminate above bad aspects of the classification method, because, true gray weights map or gray inlier corresponding to training input can be generated using the high quality template of the validation image used in training.

# Bibliography


[1] Chris Albon. *SVC Parameters When Using RBF Kernel*, 2017. https://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/ [Accessed: 10 June 2020].

[2] Preeti Gopal, Sharat Chandran, Imants Svalbe, and Ajit Rajwade. Low radiation tomographic reconstruction with and without template information. *Signal Processing*, page 107582, 2020.

[3] Wikipedia contributors. Plagiarism — Wikipedia, the free encyclopedia, 2004. URL https://en.wikipedia.org/wiki/Convolutional_neural_network. [Online; accessed 12-June-2020].