



Indian Institute of Technology, Bombay

Particle Picking in Cryogenic Electron Microscopy

Khursheed Ali

MTech (CSE)

Guidance by
Prof. Ajit Rajwade

May 5, 2018

Contents

1	Overview	3
2	Background	4
2.1	Tomographic Reconstruction	4
2.1.1	Introduction	4
2.1.2	Tomographic Projection	5
2.1.3	Fourier Slice Theorem	5
2.1.4	Filtered Back Projection (FBP)	6
2.1.5	Reconstruction using Compressed Sensing	7
2.1.6	Application	7
2.2	Tomography under unknown angle	7
2.2.1	Moment Based Reconstruction	8
2.2.2	Order Based Reconstruction	9
3	Introduction to Cryo-EM	10
3.1	Motivation	10
3.1.1	X-ray Crystallography	10
3.1.2	Cryogenic Electron Microscopy (Cryo-EM)	11
3.2	Basic Pipeline	12
3.2.1	Particle Picking	12
3.2.2	CTF correction	14
3.2.3	Clustering	14
3.2.4	Angle Assignment	15
3.2.5	3D Reconstruction	15
3.3	Challenges	16
3.3.1	Radiation damage	16
3.3.2	Noise	16
3.3.3	Unknown angles	17
3.3.4	Heterogeneity	17
3.3.5	Manual Particle Picking	17
4	Particle Picking using Machine Learning	18
4.1	Semi-Automated Method	18
4.2	Dataset: EM Databank	18

4.3	Generation of Synthetic Data	19
4.3.1	Parallel Beam Projection	19
4.3.2	Micrograph Generation	20
4.3.3	Generating Positive and Negative sample	22
4.4	Training Architecture	23
4.5	Particle Detector Architecture	24
4.6	Experiment	25
4.7	Feature Extraction	26
4.8	Support vector machine (SVM)	27
4.8.1	Results	28
5	Future work	33

Chapter 1

Overview

Cryogenic Electron Microscopy (Cryo-EM) is a special type of transmission electron microscopy (TEM) which is used for getting projections (tomographic projections) for very tiny biological sample like bacteria, protein etc. in cryogenic environment. Then, these projections are used for creating 3D internal structure of particles. But output from cryo-em is a very large 2D image (also called as micrograph) having lots of projection. So before going for 3D reconstruction, particles projections are to be marked out manually or automatically for reconstruction. Marking of particles is called as *particle picking*.

Micrograph contains thousands of projections, so marking manually is very time consuming process. Also, 3D reconstruction requires projections to be aligned on their center for robust reconstruction. So marking thousands particles manually with very high precision is very time consuming. Because of that, there is a need of robust automatic algorithm which can mark particles and with high precision.

Chapter 2

Background

2.1 Tomographic Reconstruction

2.1.1 Introduction

The objective of tomographic reconstruction is to obtain 3D representation of the internal structure of 3D object. For example, in computed tomography (CT) scan of human head (figure 2.1), the structure of the brain can be seen.

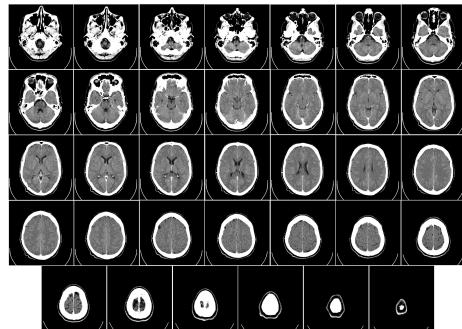


Figure 2.1: Computer tomography of human brain, slice of 3D volume from base of the skull to top. Source: wiki

Before creating 3D internal structure of the object, there are in between many steps which have to be performed. In order to make it simple, the example of CT machine can be considered. In CT machine, X-rays are used because it has ability to penetrate most of the objects. Using X-rays, many 2D projections are taken for the 3D object (in case of 2D object, 1D projections are taken) at different angles. These projections are also called as *tomographic projections*. These tomographic projection are defined as *Radon transform*. Then, by using all these 2D tomographic projections, internal structure of 3D object is created.

2.1.2 Tomographic Projection

For better understanding, consider 2D object and it's 1D projection (figure 2.2) as it is much easier to understand. But, this concept can easily be extended for 3D objects and their 2D projections. For taking projection, a parallel beam of X-rays are fired on the object at some fixed angle θ_k . The degree of absorption of X-rays by the object are recorded by the detector which is called as projection at angle θ_k . Now mathematical equation for the projection can be formulated. Let 2D object be defined as $f(x, y)$. Then Projection at angle θ_k is given by

$$R(f) = g(\rho_j, \theta_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta_k + y \sin \theta_k - \rho_j) dx dy \quad (2.1)$$

$R(f)$ is called as *Radon transform* of function f . $g(\rho_j, \theta_k)$ is read as projection of $f(x, y)$ at angle θ_k and ρ_j distance from origin.

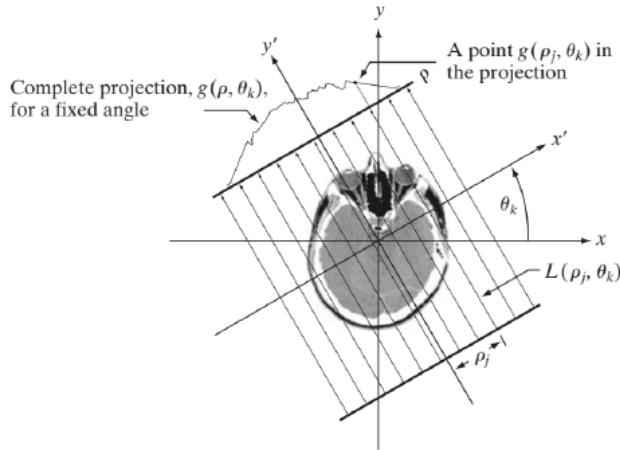


Figure 2.2: Parallel Beam (X-ray) Projection.
Source: Rafael C. Gonzalez , Richard E. Woods, Digital Image Processing

2.1.3 Fourier Slice Theorem

Fourier Slice Theorem gives us the relationship between Fourier transform and Radon transform. Radon transform of $f(x, y)$ is by eq. (2.1). The 1D Fourier transform $g(\rho, \theta)$ w.r.t ρ with fixed θ is given by eq (2.2).

$$G(\mu, \theta) = \int_{-\infty}^{\infty} g(\rho, \theta) e^{-j2\pi\mu\rho} d\rho \quad (2.2)$$

Now expanding the $g(\rho, \theta)$ and simplifying the equation

$$G(\mu, \theta) = \int_{-\infty}^{\infty} g(\rho, \theta) e^{-j2\pi\mu\rho} d\rho \quad (2.3)$$

$$G(\mu, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\mu\rho} dx dy d\rho \quad (2.4)$$

$$G(\mu, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \left[\int_{-\infty}^{\infty} \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\mu\rho} d\rho \right] dx dy \quad (2.5)$$

$$G(\mu, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi\mu(x \cos \theta + y \sin \theta)} dx dy \quad (2.6)$$

$$G(\mu, \theta) = [F(u, v)]_{u=\mu \cos \theta, v=\mu \sin \theta} \quad (2.7)$$

Here, $F(u, v)$ represents the 2D discrete Fourier transform of function $f(x, y)$. By eq (2.7) it can be concluded that, Fourier transform of projection of 2D object along some direction θ i.e $G(\mu, \theta)$ is equal to a slice of 2D Fourier transform of object along same direction θ in frequency plane, passing through origin. This is called as *Fourier Slice Theorem* or *Projection Theorem*. Similar theorem is also there for 3D object.

2.1.4 Filtered Back Projection (FBP)

From section (2.1.2) and (2.1.3), it can be known how projections (tomographic projections) are taken and it's relation with Fourier transform of object. By exploiting these facts, the internal structure of the object can be reconstructed. Let us consider 2D object is denoted as $f(x, y)$ and it's 2D F.T (Fourier transform) is represented as $F(u, v)$.

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(xu+yv)} du dv \quad (2.8)$$

Let $u = \mu \cos \theta$ $v = \mu \sin \theta$,

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} F(\mu \cos \theta, \mu \sin \theta) e^{j2\pi(x\mu \cos \theta + y\mu \sin \theta)} \mu d\mu d\theta \quad (2.9)$$

By using Fourier slice theorem eq (2.7) we get,

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} G(\mu, \theta) e^{j2\pi(x\mu \cos \theta + y\mu \sin \theta)} \mu d\mu d\theta \quad (2.10)$$

After simplification we get,

$$f(x, y) = \int_0^{\pi} \int_0^{\infty} G(\mu, \theta) e^{j2\pi(x \cos \theta + y \sin \theta)} |\mu| d\mu d\theta \quad (2.11)$$

So eq (2.11) says that, if there are many projection at different angle then it can recover $f(x, y)$ i.e object internal structure from projections. But, one concern is there i.e in eq (2.11) $|\mu|$ (ramp filter) is a *unbounded* function, therefore it is not integrable. In order to solve this problem Ram-Lak filter ($\text{rect}(\mu D)$) or Ram-Lak hamming filter is used. Updated FBP equation is given by eq (2.12) using Ram-Lak filter.

$$f(x, y) = \int_0^\pi \int_0^\infty |\mu| \text{rect}(\mu D) G(\mu, \theta) e^{j2\pi(x \cos \theta + y \sin \theta)} d\mu d\theta \quad (2.12)$$

2.1.5 Reconstruction using Compressed Sensing

In this section, another method for reconstruction using the set of projections taken at different angles is discussed. This method uses the concept of Compressed sensing (CS). CS works here because number of angles of projection are limited due to various reason like cost, energy and health considerations. This type of problem is also called as *angle starved* problem. It is known that images are sparse or compressible in standard basis like DCT. So CS based optimization equation for tomographic reconstruction is give by eq (2.13)

$$E(\beta) = \|\mathbf{y} - \mathbf{R}\mathbf{U}\beta\|^2 + \lambda \|\beta\|_1 \quad (2.13)$$

Here, \mathbf{y} vector is created by concatenating 1D projections of various angle. \mathbf{U} is the basis matrix in which image/object $f(x, y)$ is sparse or compressible. β is $f(x, y)$ representation in \mathbf{U} basis i.e $f(x, y)$ satisfy eq (2.14). \mathbf{R} is the Radon matrix (it can also be Radon operator)

$$\text{vec}(f)) = \mathbf{U}\beta \quad (2.14)$$

Eq (2.13) can easily be solved using many optimization algorithm such as ISTA. It is being observed that when number of angles are less then CS based reconstruction gives good result as compared to FBP up to certain limit.

2.1.6 Application

Tomographic reconstruction has many application mostly is health care such as CT scan, industrial application such as fault detection in machine, observation of plant roots, remote sensing i.e observing underground objects or phenomena and most importantly study of protein, bacteria, Ribosome, cells and virus by biologist for creating medicine or vaccination or for any other purpose.

2.2 Tomography under unknown angle

In previous section (2.1), reconstruction from projections was seen where angle at which each projections were taken were known. But if only projections are

known but not angles then, previously seen method for reconstruction will not be applicable.

There are various application where angles are not known such as patient moving during CT scanning (this mostly happens if patient is baby), machine fault, moving insect tomography, cryo-electron tomography and many more. In all these case, only projections are there but not angles, this case is called as *tomography under unknown angle*.

This section talks about reconstruction of internal structure of object back from projection when angles are not known i.e. only with the help of tomographic projections.

2.2.1 Moment Based Reconstruction

Again, for simple understanding lets take 2D object and it's 1D projection. But this logic can easily be extend for 3D object and it's 2D projection.

Let say $f(x,y)$ be the object then, moment of order (p,q) for $f(x,y)$ is given by eq (2.15).

$$M_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) x^p y^q dx dy \quad (2.15)$$

Let $P_\theta(s)$ be the projection at angle θ then moment of order (n) is given by eq (2.16)

$$M_\theta^{(n)} = \int_{-\infty}^{\infty} P_\theta(s) s^n ds \quad (2.16)$$

where,

$$P_\theta(s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \delta(x \cos \theta + y \sin \theta - s) dx dy \quad (2.17)$$

Therefore, by substituting $P_\theta(s)$ in eq(2.16) by eq(2.17) and after simplification,

$$M_\theta^{(n)} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) (x \cos \theta + y \sin \theta)^n dx dy \quad (2.18)$$

So, the Helgasson Ludwig Consistency Conditions (HLCC) [5] gives us the relation between the image moment and its projection moment and which is given by eq(2.19).

$$M_\theta^{(n)} = \sum_{l=0}^n f(x,y) C_n^{n-l} (\cos \theta)^{n-l} (\sin \theta)^l M_{n-l,l} \quad (2.19)$$

Here, projections of the $f(x,y)$ is known. So it's moment i.e $M_\theta^{(n)}$. By eq(2.19) unknown angles can be found out by iteratively solving for angles [4]. Once the angles for these projections are found then any of the earlier mentioned method for reconstruction can be applied.

Advantages of this method are simple to implement, makes direct use of consistency conditions and work well for small number of angles. But, major drawback of this method is that it is highly sensitive to noise.

2.2.2 Order Based Reconstruction

This method is very much similar to machine learning based method. In this, all acquired projections are sorted i.e. in order of increasing angles. Only issue is that this method requires large number of projection and it assumes the all unknown projection angles are independently sampled from uniform distribution (it can be from any known distribution but then accordingly algorithm will change). Assuming distribution to be uniform then this problems reduce to matching problem where we have to match each projection to one of the angles sampled evenly from unit circle.

It's a iterative algorithm, where initial projection say P_0 and one angle say θ_0 are chosen. Without lose of generality, that projection is mapped with angle say θ_0 . Then, it finds closest projection to P_0 and then that projection is mapped with next angle from sorted distribution. This process is repeated till full mapping is completed.

Once it finds the angles for all projections then apply any of the earlier mentioned method for reconstruction.

Chapter 3

Introduction to Cryo-EM

3.1 Motivation

There are many applications where tomography concept is being used such as CT in which X-Rays are used. Suppose biologist wants internal 3D model of a protein particle. Then, for this type of problem, CT machine cannot be used because it is not designed for it nor a light-microscope because it is designed for viewing surface. Since late 20th century, a method called as *X-ray Crystallography* is being used for this kind of problem, when sample (object under study) is very tiny[8].

Next section talks briefly about *X-ray Crystallography* and problem faced in it. Also the benefit of *Cryo-EM* over X-ray Crystallography.

3.1.1 X-ray Crystallography

As the name suggest, X-ray is being used for taking projections of tiny particle like protein or DNA. It uses the Braggs's law of *X-ray diffraction* by crystal for taking the tomographic projection. Here, in figure (3.1) DNA sample is firstly crystalline and then placed in between focused X-ray beam and detector. Because of crystal, X-rays are diffracted and this diffraction is captured on the detector screen. Then from the recorded reading model reconstruction is done.

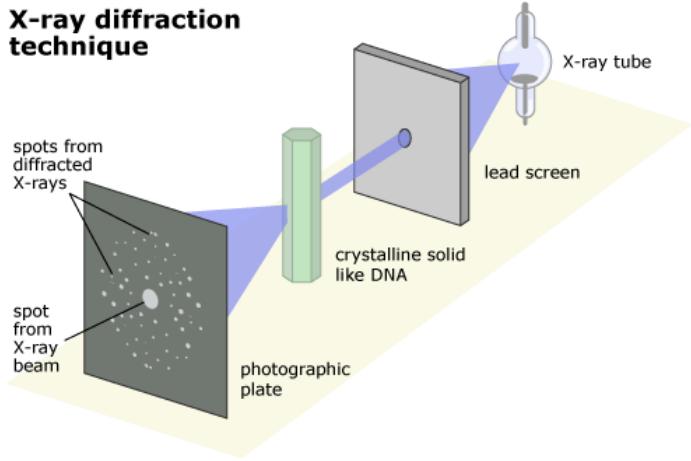


Figure 3.1: X-ray Diffraction.
Source: undsci.berkeley.edu - article-dna 04

3.1.2 Cryogenic Electron Microscopy (Cryo-EM)

Cryo-EM uses the electron beam for taking projections. Because of that, resolution of cryo-em is much better than X-ray crystallography. Depending on the type of sample under study, power of electron dose is set to get maximum possible resolution for that sample [3]. Here, the procedure for sample creation is completely different than X-ray crystallography. Firstly, a slice of sample to study is placed in vitrified water (water with some amount of ethane) then freeze below -150°C [3] [8]. After that slide is placed in electron microscopy (EM) and observation/projection are made in cryogenic conditions. As whole thing is carried out in cryogenic conditions and appliance used is Electron-Microscope (EM) that's why this is called *Cryo-EM*. See figure (3.2) for working of Cryo-EM.

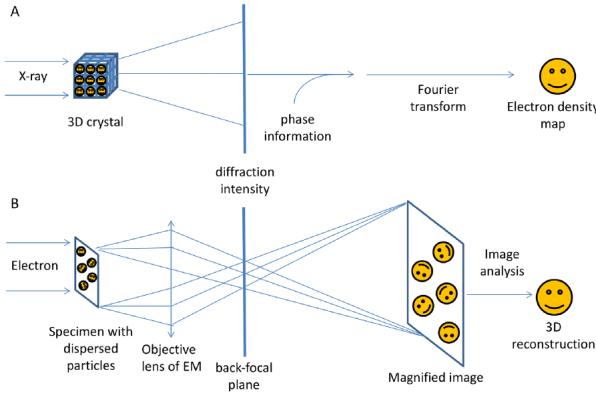


Figure 3.2: Functional difference between X-ray crystallography (A) and single particle cryo-EM (B) Source: Wang et al. [8]

X-ray crystallography resolves structures of macro-molecules at atomic level. But, cryo-em can provide structural information at resolution from 3\AA to 3nm . As in cryo-em sample creation doesn't want sample to be in crystalline form so it can also examine non-crystalline structure. Also, it reveals the structure close to its native state than X-ray crystallography [8]. See figure (3.2) for working difference of both methods.

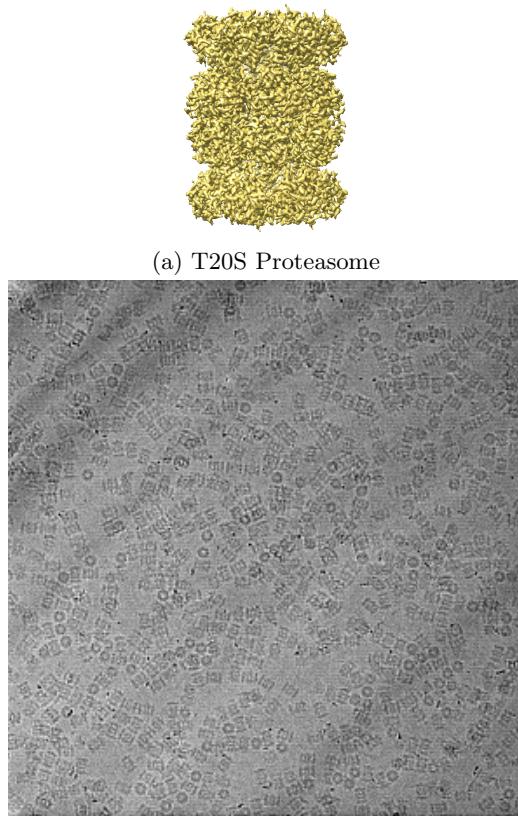
3.2 Basic Pipeline

This section talks about cryo-em whole base pipeline starting from selecting the particle from micrograph (image (b) 3.3) also called as *particle picking* generated from Cryo-EM to it's 3D model reconstruction (similar to image (a) 3.3).

3.2.1 Particle Picking

Cryo-EM gives micrograph of very large size example 7000×7000 . These micrograph contain lots of radon projections of the particles (around in thousands) taken by parallel beam because EM shoots electron beam in parallel. For reconstruction process, the very first steps is to manually or automatically mark out these projections by drawing bounding box around particles or just by marking its center in micrograph (3.4) and then crop out these projections. After that all the cropped out projections are collected into *stack* of projection images. As micrograph is very huge and it contains large amount of projections, so manually marking is very time consuming process. Now a days, biologist crowds source this initial phase of particle picking [1]. But, even after crowdsourcing, they have to remove wrongly marked particle by themselves which is still a little time consuming.

¹EBI data bank: <http://www.ebi.ac.uk/pdbe/empdb/empiar/entry/10025/>



(b) Micrograph of T20S Proteasome at 2.8 Å Resolution
Source: EMPIAR-EBI data-bank

Figure 3.3: Source: EMPIAR-10025 EBI data-bank: <https://bit.ly/2JrFCD5>

For saving time in particle picking, there are few automated and semi-automated algorithms. Example for one such automated algorithm is "difference of Gaussian" where a broad 2D Gaussian function (i.e. having large σ) convolve with micrograph is subtracted from narrow 2D Gaussian function (i.e. having smaller σ) convolve with micrograph. Result of this method gives circular white object pattern with a dark surrounding [7]. Problem with this approach is that tuning of hyperparameters like here σ depending on the type particle and noise level in micrograph. An example of semi-automated method is a machine learning based particle picking [2]. In case of machine learning, machine learning model will learn to detect particle in same noise level environment as of micrograph because training samples are taken from these micrograph itself. The next chapter talks about some semi-automated method and new detector design and it's experimental results on simulated data.

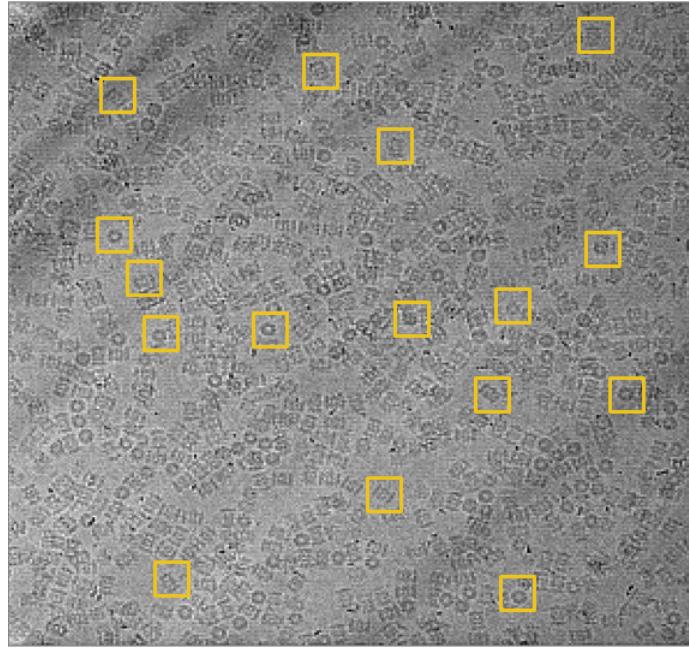


Figure 3.4: Manual particle picking from micrograph of T20S Proteasome
Source: EMPIAR-10025 EBI data-bank¹

3.2.2 CTF correction

Image formation in Cryo-EM for biological samples taken under cryogenic environment by freezing the sample in vitrified water can mathematically be described as Contrast Transfer Function (CTF). CTF depends on many parameters such as defocus, astigmatism, beam coherence, energy spread and tilting of the sample [3]. CTF oscillates rapidly, with some Fourier components transferred with positive contrast and other with inverted contrast. CTF is directly proportional to defocus i.e. if defocus increases then density of reversal increases which means higher contrast. So if defocus is higher, it helps in visualizing micrograph which further helps in marking of the particle and determining its orientation. But, this high defocus give rise to astigmatism i.e circular ring in CTF turns into ellipse [7]. Because of all these issues, *CTF correction* is needed for the micrograph.

3.2.3 Clustering

The projection image from the stack of images is highly noisy i.e it has very low SNR value since the electron beam used for taking projection is of very low power otherwise it will damage the sample. Also, distortion due to CTF effects makes it difficult to evaluate the projection. So, in order to resolve this

problem, *clustering* of all projections from stack are done. After the clustering, all projections within a cluster are averaged out to get one average projection per cluster. This helps a lot in reducing the noise.

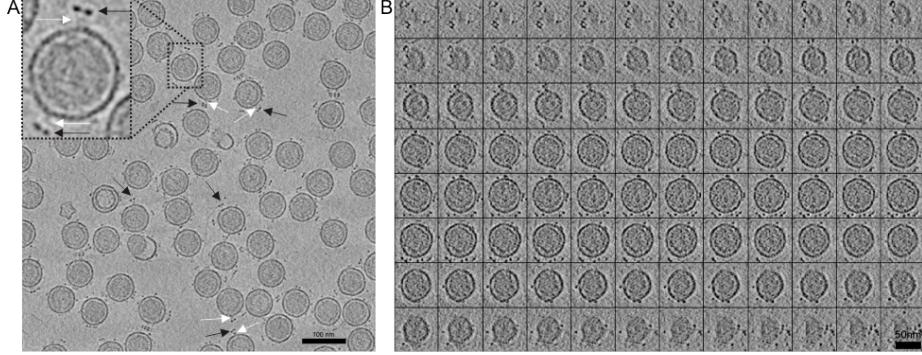


Figure 3.5: a) Phi12 bacteriophage micrograph b) Each row represent similar projection. Showing cluster of of projection before averaging.

Source: med.nyu.edu²

There are lots of method for performing clustering. Algorithm used by RELION (REgularized LIkelihood OptimizatioN) uses maximum likelihood 2D classification, its a most widely used software tool which performs all the base pipeline task [6]. In brief, this algorithm firstly select an initial set of reference projections. Then, for each projection, probability w.r.t rotation and translation with each reference projection is calculated. Then averaging is carried out for projections with higher probabilities. Averaged projections are then taken as new reference projections and whole process is repeated few times.

3.2.4 Angle Assignment

After clustering (section 3.2.3) step, projections are little noise free and CTF correction has been done up to certain level because of averaging. Now, 3D model is reconstructed using these projections, but for 3D reconstruction there is a need of angles at which these projections are taken if reconstruction is done using method discussed previous section i.e FBP (2.1.4) or CS (2.1.5). Finding angles and then reconstruction is called as tomography under unknown angle. So angles can be found out using two methods i.e. Moment-based (2.2.1) and Order-based (2.2.2).

3.2.5 3D Reconstruction

Once angles for each projection are known then 3D reconstruction is possible using FBP (2.1.4) or CS (2.1.5) based method. As always, if there are variety

²EMPIAR-1002 data bank: <https://med.nyu.edu/skirball-lab/stokeslab/phi12.html>

projections at different angles then 3D reconstruction will be more accurate. But in *SPR (single particle reconstruction)* [7] many times getting projections at various angles is not possible because of the particle orientation. As projection angle of EM is fixed i.e. perpendicular to slide. In EM, slide is the place where sample is placed for observation or projection. So the only way to get projections at different angles is from particles orientation in frozen vitrified water sample [3]. But problem is that particles orientation in sample is not fully random. So there will always be the case that along certain direction projection will not be acquired. For solving this problem, biologist tilt the sample slide with some angle and then projection are taken. This tilting is done many times in many direction to cover large possible directions. If projections are taken this way, then effect of CTF will be different for different tilt direction and also within slide because of tilt. In this type of situation CTF correction is more challenging [3]. This method is called as *Cryo-EM Tomography*.

3.3 Challenges

3.3.1 Radiation damage

In Cryo-EM, electron beam is used. Problem with it is that these electron start interacting with the sample electron. In other terms, if the exposure time of vitrified biological sample is increased then this leads to breaking of covalent bonds of sample (because of electron-electron integration) and progressively it will do the structural degradation[3]. Therefore, total dose of electron that can be used is limited, this also affects the resolution i.e. magnification of cryo-em which means depending on the biological sample under study resolution will be different.

3.3.2 Noise

Higher power electron dose usually destroys the sample, so for handling these kind of issues, low power electron doses are used. But, this leads to increase in noise w.r.t. to signal i.e. SNR for the micrograph become very low. In Cryo-EM, the distribution of noise is *Poisson distribution* defined by eq (3.1), which makes the problem more challenging as Poisson noise is signal dependent. In case of Image $I(x,y)$, it is given by equation eq (3.2) which means removal of Poisson noise is far more challenging than signal independent noise like (i.i.d Gaussian noise)

$$P(X = i) = \frac{\lambda^i}{i!} e^{-\lambda} \quad (3.1)$$

$$Y(x,y) \sim \text{Poisson}(I(x,y)) \quad (3.2)$$

$Y(x,y)$ represent Poisson-noisy version of $I(x,y)$.

3.3.3 Unknown angles

When projections are taken in cryo-em, angle at which beam is fired is known, but problem here is that when biological samples are frozen in vitrified water, then orientation of these samples are random. So, even if the angel at which beam is fired is known, the projection angles are different for every particle because of their orientation. As orientation of these samples are random which also means projection angles are also unknown, this information lose causes problem at the time of reconstruction from these projections.

3.3.4 Heterogeneity

Sample of same biological particles many not be identical i.e. every particle has some heterogeneity associated with it. Because of this complexity in whole process increases. At the time of selecting particle from micrograph, one has to make sure that consistency of particle has been marked otherwise this causes problem at the time of finding angles and also it may give rise to wrong 3D structure reconstruction [7].

3.3.5 Manual Particle Picking

Picking of particle is mostly time consuming and an important task. As discussed previously, wrong particle picking will lead to wrong result. Still today, many prefer to mark these particle in micrograph with hands so that error in particle picking can be reduced.

So, by the nature of problem, there is a need of robust automatic or semi-automatic algorithm for particle picking which can handle outlier like few wrongly marked particle. Also these algorithm should be fast at the time of finding the particle location in micrograph. The next chapter is mostly focused on first stage of the pipeline i.e. *particle picking* and will talk about few semi-automated methods and their testing on simulated data set.

Chapter 4

Particle Picking using Machine Learning

4.1 Semi-Automated Method

In semi-automated procedure, there is some need of human intervention in the procedure. This chapter talks about such type of semi-automated method for particle picking. All methods discussed here require only one time human intervention and that even at the start of method such as training of *Support Vector Machine (SVM)* classifier. Here, human will mark out few projections manually on micrographs say around 2K (far less than 30K) and then these projections are given as input to machine learning based classification algorithm which then try to distinguish between particle and background.

4.2 Dataset: EM Databank

EM Databank¹ is globally recognized portal where one can deposit or retrieve the 3D density map, atomic map and their meta-data information. Some other datasets which are part of this set are EBI, PDB (Protein Data bank), PDEe and many more. It also has variety of software tools such as for viewing and analyzing atomic models, parsing of .mrc or .map file format. It also has micrographs for most of the particles. In this section, for simulation purpose three 3D datasets EM-2211 (4.1a), EM-5689 (4.1b) and EM-5693 (4.1c) have been used.

Generating synthetic micrograph of sample particle is similar to cryo-em micrograph. Then from these micrograph few projections of the particles are marked out for training. As simulation is being done at software level (i.e real EM is not being used) so getting micrograph from 3D model directly is challenging. So this is achieved by first taking many single projections and then merging these in grid or random pattern to get one completed micrograph. By

¹EM-Databank <http://www.emdatabank.org/>

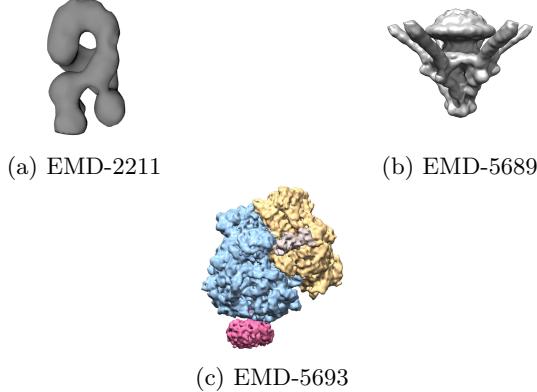


Figure 4.1: Source: EM Databank

this way, 400 micrographs (200 for training and 200 for testing) are generated for each of the three particles i.e. EM-2211 (4.1a), EM-5689 (4.1b), and EM-5693 (4.1c).

4.3 Generation of Synthetic Data

In next the section i.e. (4.3.1) and (4.3.2), talks about micrograph generation and section (4.3.2) talks about how negative and positive samples generation for training and validation

4.3.1 Parallel Beam Projection

First step in micrograph generation is to take lots of projections of the particle. In our experiment, per particle 2000 (2K) projection are taken. Note that, all three particles projection should not to be mixed i.e these are separate three independent experiments. The three particles are chosen very carefully depending upon their 3D structure i.e. EMD-5693 is some what spherical, EMD-5689 is semi-symmetric and EMD-2211 is irregular w.r.t. to different view point. Hence this represents three different types of experiments. Our anticipation is that the training will be easier in the first case (EMD-5693) as projection images under different viewing directions will be quite similar. In the third case (EMD-2211), there is considerable variety in the structural properties of the projection images at different angles. The second dataset(EMD-5689) represents a case that is intermediate between the first and third datasets.

For taking 3D projection in Matlab, Astra toolbox² is used. This toolbox supports variety of projections i.e. 2D/3D model projections, parallel beam,

² Astra toolbox: <https://www.astra-toolbox.com/>

fan beam and cone beam type of projection methods. As EM uses the parallel electron beam, so here parallel beam is used for projecting 3D model, which uses GPU power for computation. By parallel beam projection method, 2K projection are generated at equally line spaced angle θ (angle with Y-axis) and ϕ (angle with Z-axis) for particles EM-2211 (4.2), EM-5689 (4.3), and EM-5693 (4.4).



Figure 4.2: EM-2211 sample projections (dimension: 98x98)



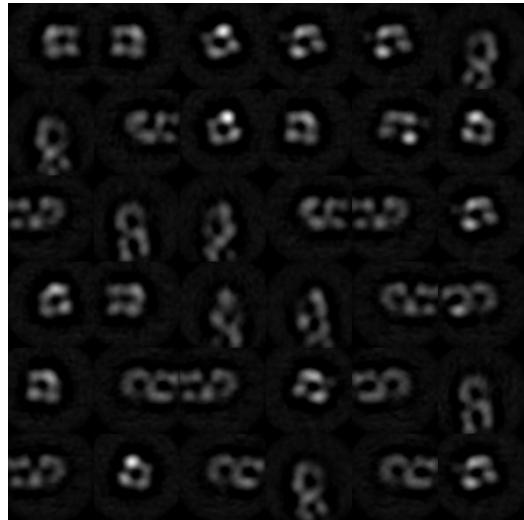
Figure 4.3: EM-5689 sample projections (dimension: 278x278)



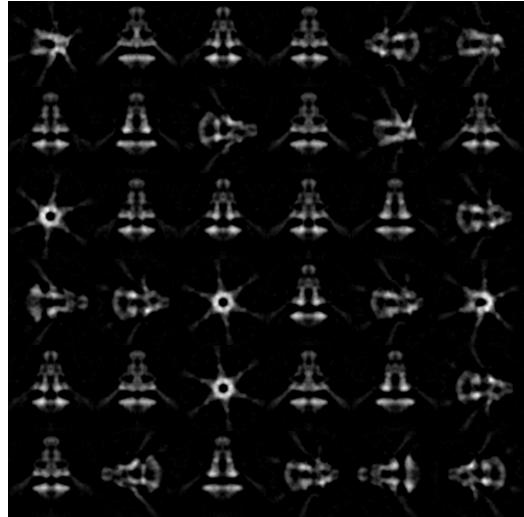
Figure 4.4: EM-5693 sample projections (dimension: 333x333)

4.3.2 Micrograph Generation

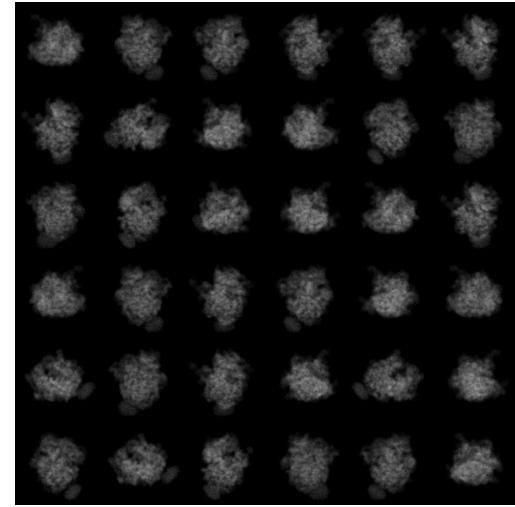
Once projections are generated then from these projection micrographs are generated with initially zero percent noise (after that different levels of Poisson noise affected micrographs are generated). Micrograph can be created in grid pattern i.e. each randomly chosen projection taken from stack of projection is placed at grid location or another way is to place at random location on very big blank image (say of dimension 2000x2000). In our current experiment, grid pattern is used but in future, random pattern will be used i.e. moving more close to real micrograph. Sample micrograph for particles EM-2211 (4.5a), EM-5689 (4.5b), and EM-5693 (4.5c) have been shown.



(a) EMD-2211 micrograph (dimension 588x588)



(b) EMD-5689 micrograph (dimension 1668x1668)



(c) EMD-5693 micrograph (dimension 1998x1998)

Figure 4.5: Micrograph

4.3.3 Generating Positive and Negative sample

In machine learning algorithm, train and test dataset are required in training and testing phase respectively. So for creating training and test sets, micrographs are used. Also, both the sets required positive and negative projections. Here, in our case, positive samples are just projections for example in case of particle EMD-5693 positive image dimension are 333x333. But negative samples includes everything except correct projection such as background, mixture of particles in one image, incomplete particle, other types of particles and outliers. So, by using particle micrograph, negative sample images are generated such that they do not capture positive case. Examples of positive samples for particle EM-2211 (4.2), EM-5689 (4.3), and EM-5693 (4.4) are shown and examples for negative samples for particle EM-2211 (4.6), EM-5689 (4.7), and EM-5693 (4.8) are shown.



Figure 4.6: EM-2211 Negative sample (dimension: 98x98)



Figure 4.7: EM-5689 Negative sample (dimension: 278x278)



Figure 4.8: EM-5693 Negative sample (dimension: 333x333)

4.4 Training Architecture

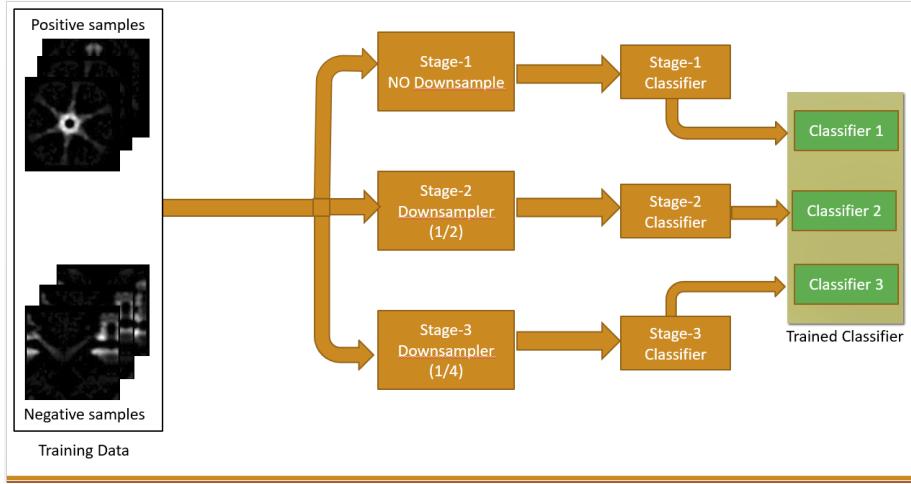


Figure 4.9: Particle Detector Training Architecture

One simple way to train machine learning classifier like SVM is to feed the **train** set in SVM classifier and tune the hyper-parameters which are best suited for the application. Then, use that trained classifier for detection purpose. But, this architecture of training will not work in cryo-em micrograph because actual micrograph dimensions are in the thousands example 7000x7000 or much bigger than this. In our experiment, dimension of micrograph of EMD-5893 is 1998x1998 (2000^2 pixels) which is largest w.r.t. other two particle micrograph. So if this simple architecture is used then time required for creating *probability map* (some also calls it *score map* or *heat map*) for micrograph will be huge. Probability of a patch centered at \mathbf{x}_i (x_i, y_i) for a positive class is calculated using per class score generated by SVM (our case only classes) and softmax function. For two class, softmax is given by eq(4.1).

$$P(y = +ve | \mathbf{x}_i) = \frac{e^{si_{+ve}}}{e^{si_{+ve}} + e^{si_{-ve}}} \quad (4.1)$$

Here, si_{+ve} and si_{-ve} are the class scores generated by SVM model for patch centered at x_i .

For example, let take micrograph of dimension 2000×2000 i.e. 4×10^6 pixels. Let say already trained classifier takes 0.001 sec for classifying one patch of dimension 333x333 centered at i th pixel and give its probabilistic score. Then for finding score at every pixel will take around 40000 secs ($4 \times 10^6 \times 0.001$) which is approximately equals to 11hrs which means that this a very poor particle picker as expected time should be in seconds or up to minutes but not in hours.

In order to overcome this problem, our new architecture (by tweaking the current way of training) follows downscaling of model i.e. in case of *3-downscale* model three classifiers are trained. First classifier is simple one i.e. no downscaling, second model train classifier by downscaling micrograph by 2 and third model at downscale of 4. So, after whole training process in completed output will be *3-trained-classifiers* in case of 3-downscale model. Figure (4.9) shows the training architecture of 3-downscale model.

Depending on the size of the micrograph, our architecture code can be increased or decreased (just by changing configuration file) considering computation time and accuracy of the system. In our experiment, 3-downscale model has been used for all the particles.

4.5 Particle Detector Architecture

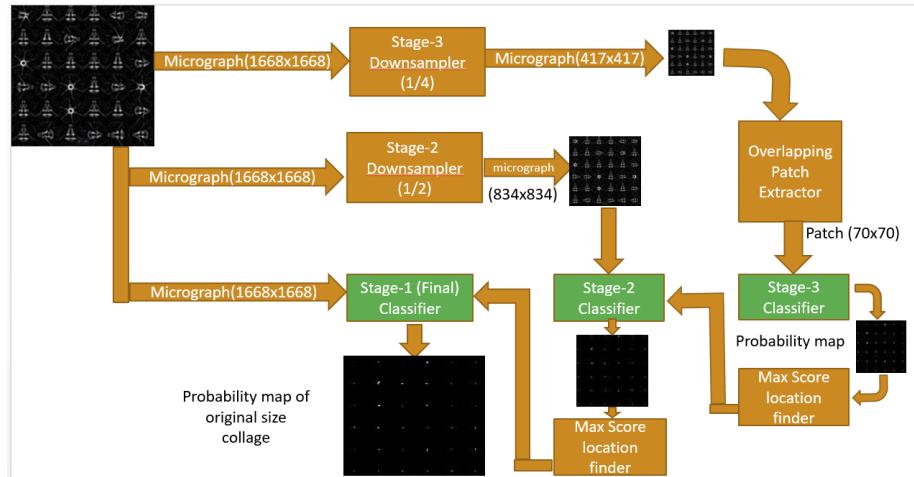


Figure 4.10: Particle Detector

Detecting particle in the micrograph and creating its probability map will take huge amount of time in case of single classifier, see section (4.4) for its computation time. So to reduce this detection time, new training architecture has been proposed i.e. *downscale* model. Detection using this new trained model is little complex as compared to a *new trained modelled architecture*. Figure (4.10) describe the work flow of detection process.

Detection process is reverse training process in terms of stages. In case 3-downscale model, first stage-3 is performed then stage-2 and at last stage-1 i.e. in reverse order. Stage-3 has one only input i.e. original micrograph is

-	Single CPU	12-CPU Pool	GPU
Our Arch Time	~ 1.15 hrs	~ 9 mins	~ 4 mins
Simple Arch Time	Not worth computing	16.97 hrs (61100 sec)	-

Table 4.1: Computation Table

downsampled by $(1/4)$ and remaining $i-1$ stage will have 2 inputs i.e. for stage- j ($j < 3$) one input will be micrograph downsampled by $1/2^{j-1}$ and other input will be the output of $(j+1)$ th stage passed through "Max score location finder" function. "Max score location finder" finds all the (x,y) coordinates in the probability map which are having probability values more than certain *min-threshold* (default 0.8). Min-Threshold has to be set depending on the how well a classifier is being trained. In our experiments, it has been set to 0.75. Apart from n th stage (here 3rd) remaining stages do not run on every pixel. They use the location given by the previous $(j+1)$ th stage. After up scaling these locations, for these location and their 4 point neighbour probabilistic score is calculated using input micrograph at that stage using that stage classifier.

Our code also supports three execution modes. These are single CPU, CPU Pooling and GPU. Below is the computational table (4.1) for getting probability map of micrograph (4.5c) of dimension of 1998x1998 three modes.

4.6 Experiment

In our experiment, three particles are under study for checking the robustness of our designed train and detector architecture with different classifiers. Also, they are tested under 7 different level of Poisson noise. Poisson noise is generated for 7 different total intensity level of micrograph. Table (4.2) shows the total intensity level at which Poisson noise is generated. Lesser the intensity level more will the effect Poisson noise, so "TI (total intensity)/500" micrograph will be highly noisy as compared to others.

-	Total Intensity (TI)	TI/2	TI/10	TI/20	TI/100	TI/200	TI/500
EM-2211	1.12e6	5.6e5	1.12e5	5.6e4	1.12e4	5.6e3	2.17e3
EM-5689	9.19e6	4.59e6	9.19e5	4.59e5	9.19e4	4.59e4	1.78e4
EM-5693	7.03e6	3.51e6	7.03e5	3.51e5	7.03e4	3.51e4	1.49e4

Table 4.2: Total Intensity at seven levels

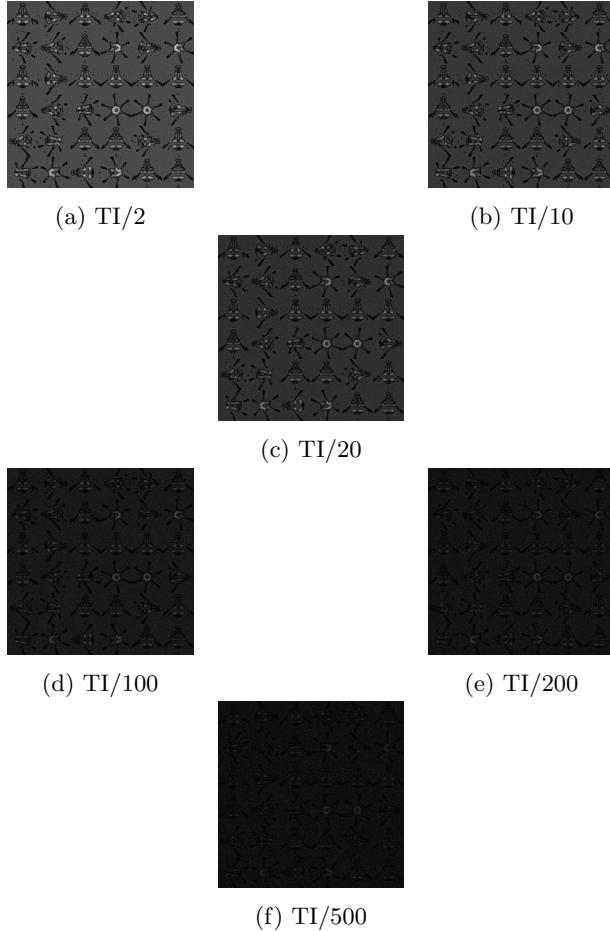


Figure 4.11: EM-5689 at 6 different noise levels (7th one is without noise, not shown here). Background intensity is set to 128.TI stands for total intensity)

4.7 Feature Extraction

As train image dimensions are very large i.e. for 333x333 image will have 110889 pixels which is a huge number if feature are considered. So to train classifier like SVM, Random forest or any other ([a part](#) from deep nets) there is a need of few feature points i.e. in few hundreds or thousands otherwise training on raw image will take many hours and resources[2].

In our experiment, PCA ([Principle](#) Component Analysis) is used for reducing the dimension of the train and test set. Number of principle component taken is 1498, so the dimension of image has reduced from 110889 to 1498 pixels. These PCA coefficients of images are then used for training the classifier. For testing

test image (say 333x333), first PCA coefficients are computed or one can say that test image is projected on to the PCA space and then it is given to detector for prediction. Because classifier is trained in PCA space, so testing has to be done in PCA space. Testing on micrograph follows (4.5) architecture, otherwise if overlapping patches are cropped on original micrograph then it will take many hours.

4.8 Support vector machine (SVM)

Support Vector machine is a machine learning algorithm used for classification. In our case, only two classes are there, one is positive class i.e. true projection and negative class. Also, as compared to deep nets it require far less data points to train.

SVM tries to optimize following objective function eq (4.3). Let y^i be the true label of \mathbf{x}^i data point. In our case, as only two classes are there, so y^i will be 1 for positive image and -1 for negative image. In eq(4.2), \mathbf{b} is bias term (unknown), $\phi(\mathbf{x})$ is a feature function i.e. given input data point \mathbf{x} it returns features vector and \mathbf{w} is weight vector which classifier will learn along with \mathbf{b} . ξ_i represent the slackness condition. In case of SVM, ξ_i is used when there is not perfect separability of the data point.

$$y^i = \mathbf{w}^T \phi(\mathbf{x}^i) + \mathbf{b} \quad (4.2)$$

$$\begin{aligned} (\mathbf{w}^*, \mathbf{b}^*) &= \underset{\mathbf{w}, \mathbf{b}}{\operatorname{argmin}} \frac{\|\mathbf{w}\|_2}{2} \\ &\text{subject to,} \\ &y^i(\mathbf{w}^T \phi(\mathbf{x}^i) + \mathbf{b}) \geq 1 - \xi_i \\ &\xi_i \geq 0 \quad \forall i = 1, 2, \dots, n \end{aligned} \quad (4.3)$$

For better classification $\phi()$ places a vital role. $\phi()$ is not learned during training, it should be known before hand. So to solve this problem, there is alternate representation of SVM classifier i.e using Kernel function $\mathbf{K}(.,.)$. Kernel representation do not use $\phi()$ and it is given by eq (4.5). In eq(4.5) y_i is the class label of the datapoint x_i , $\mathbf{K}(.,.)$ is the kernel function and C is the hyperparameter. In our experiment, RBF (Radial basis function) kernel is being used which is defined by eq(4.4).

$$\mathbf{K}(\mathbf{x}, \mathbf{x}') = e^{\frac{-\|\mathbf{x} - \mathbf{x}'\|}{2\sigma^2}} \quad (4.4)$$

$$\begin{aligned}
\max_{\alpha} \quad & \frac{-1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) + \sum_i^n \alpha_i \\
\text{subject to,} \\
\forall \alpha_i \in [0, C] \\
\sum_i^n \alpha_i y_i = 0
\end{aligned} \tag{4.5}$$

4.8.1 Results

Training and particle picking on micrograph uses 3-downscale architecture and classifier as SVM for all the three particles i.e EM-2211 (4.2), EM-5689 (4.3), and EM-5693 (4.4).

For each particle, size of train and test data set are 3450 and 2447 respectively. In train set 20% is used as validation set. Training SVM classifier took 45 to 50 mins and testing on micrograph took maximum 4mins for EM-5693 (largest micrograph w.r.t. other two) using our proposed detection method i.e see section (4.5) for detail.

Experimental observation says that while training model at different noise level, stage-1 classifier has less prediction rate than stage-3rd classifier i.e. after downsampling prediction rate is higher. It can clearly be seen from the plot (4.12,4.13,4.14) of three particles. This is due to fact that local features in large image (333x333) are distributed in large local region, so for getting good accuracy at stage-1 big train dataset will be required. But in cryo-em big train dataset means, biologist has to manually mark more projections in micrograph, which is again time consuming process.

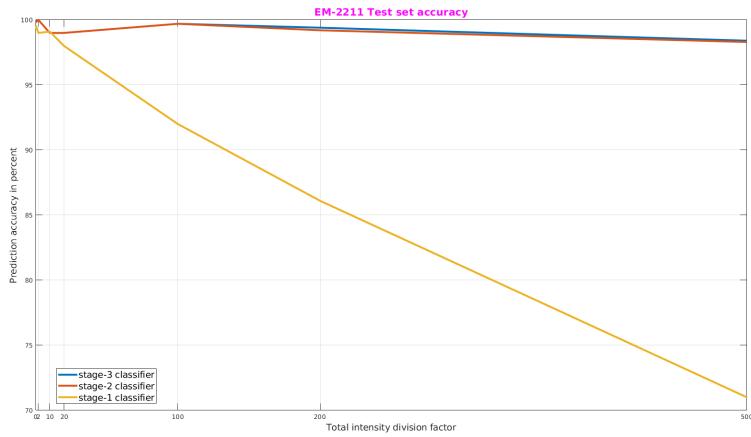


Figure 4.12: SVM classifier prediction result on test data for EM-2211

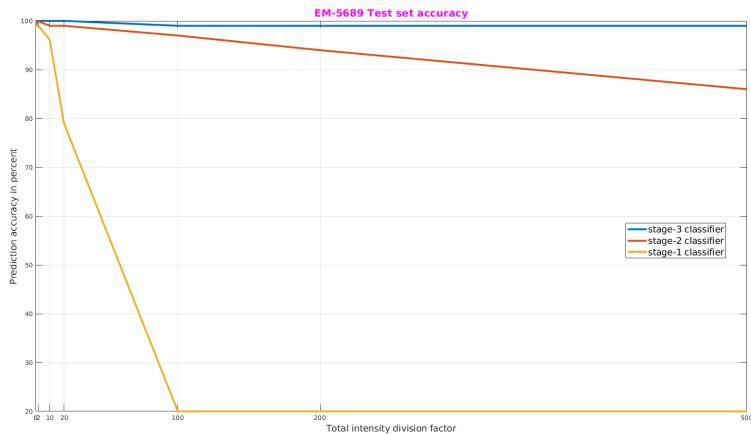


Figure 4.13: SVM classifier prediction result on test data for EM-5689

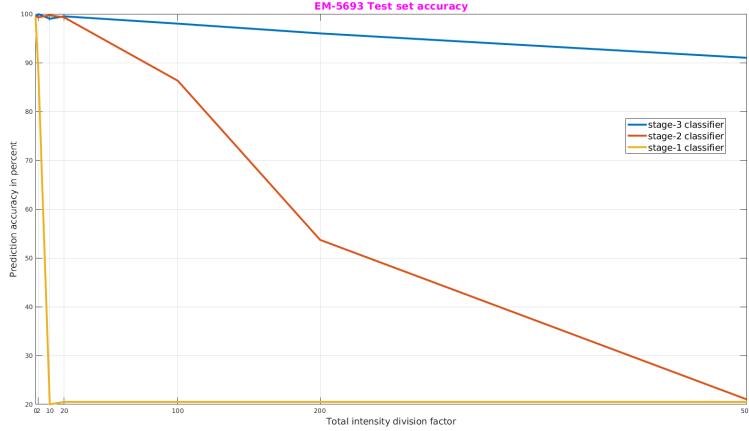


Figure 4.14: SVM classifier prediction result on test data for EM-5693

Particle picking on test micrograph using our proposed particle picker architecture (for detail see section 4.5) gives the very promising result in both terms i.e. prediction accuracy and time for simulated data. For creating the probability map for largest micrograph i.e. 1998x1998 it took maximum of ~ 4 mins using GPU. Here are few micrographs and its probability for 2 noise level (i.e. noise at 20 and 50 total intensity division factor i.e. TI/20 and TI/500)

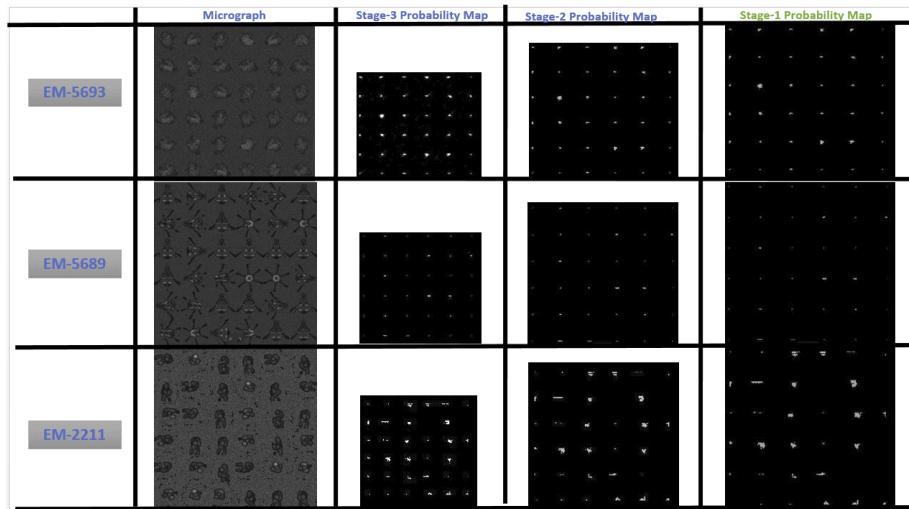


Figure 4.15: Micrograph probability map with all the three stages at noise level 'Total intensity/20'. Output of stage-1 is the final output.

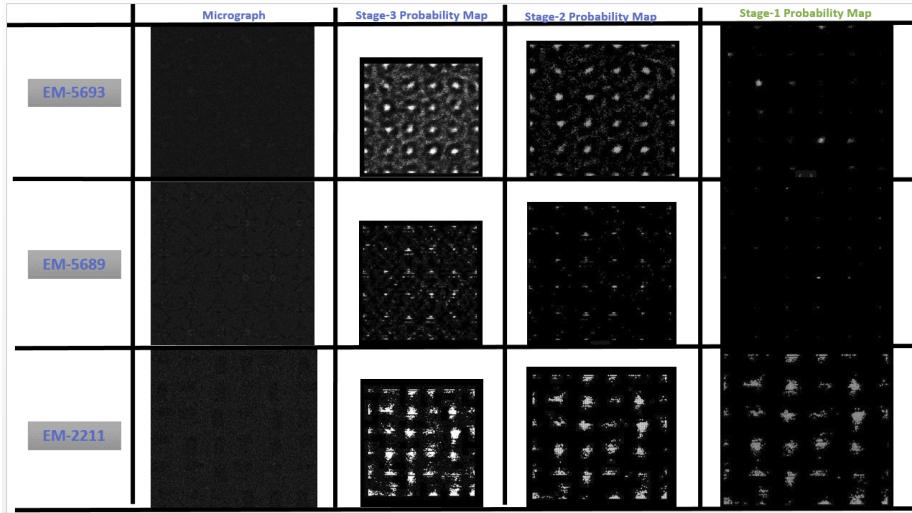


Figure 4.16: Micrograph probability map with all the three stages at noise level 'Total intensity/500'. Output of stage-1 is the final output.

Even if the stage-1 classifier prediction rate is very less. But, when all the 3 stages are combined together, then detection power of whole 3-stage-detector is increased. As one can see in figure (4.16) that at very high noise, detector is able to predict most location of particle in the micrograph very near to its ground truth location. Using the stage-1 probability map, the location (x,y) of the particle is chosen such that it has max probability (greater than some min-threshold probability) in its neighbour, this method of selection is also called as *non-maximal suppression (NMS)*. Figure (4.17) and (4.18) shows the predicted location in red using NMS and ground truth location in green.

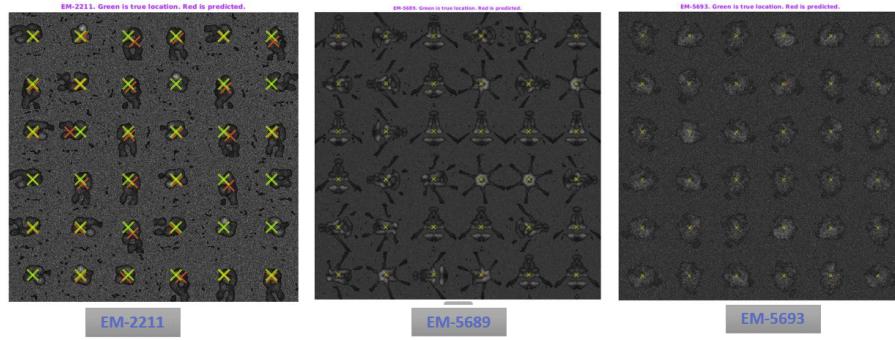


Figure 4.17: Particle picking in micrograph with noise level as 'Total intensity/20'. Green marker shows ground truth location and Red marker shows predicted location.

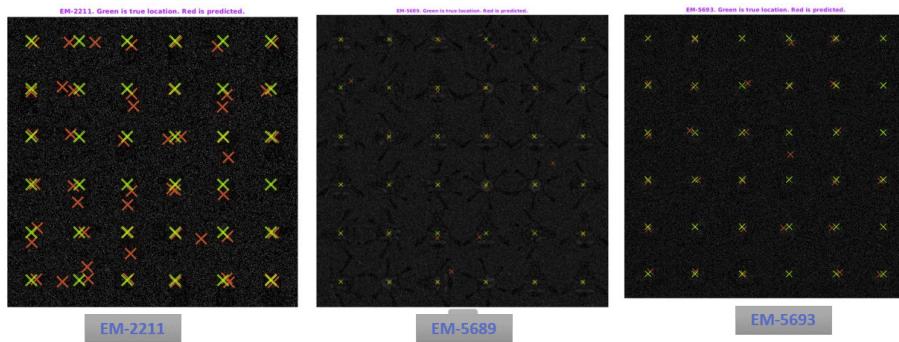


Figure 4.18: Particle picking in micrograph with noise level as 'Total intensity/20'. Green marker shows ground truth location and Red marker shows predicted location.

For our dataset, detector works pretty well up to the noise level "Total Intensity/500" with some translation error and false location. As noise level is increased, our SVM classifier starts detecting false particles. Also as the distance between predicted and ground truth location increases, it can clearly be seen in image (4.16). Reason for this false prediction is lack of training set because in case of high noise, classifier starts learning on noise rather than features as SNR of train set become very low. This also cause displacement from its ground truth location.



Chapter 5

Future work

Our experiments are tested on simulated dataset not on actual dataset. So testing on actual micrograph from cryo-em is very important. Because this will give efficiency (time and accuracy) of our model in real world scenario that will decide whether our model can be used for practical purpose or not. One problem with particle picking is translation error i.e displacement from true center. This translation error will create problem at the time of 3D reconstruction. **So for testing 3D reconstruction accuracy, third-party software like Imagic, Eman, Relion, Cryo-Sparc, etc can be used.** This will help in deciding how precisely the model is able to mark out the particles on micrograph.

Micrograph generated for biological sample from cryo-em has very low SNR value. So training classifier from very noisy data will **become** give less accuracy. In our experiments, denoising of the micrographs was not handled. So one way to handle noisy data is to first denoise the train set and then train the model.

For training classifier (except Convolution Neural Network (CNN)) directly on raw image pixels is not advisable because 100x100 image will have 10K pixels i.e features and training on 10K feature will take too much time, also large amount training data is required for correct classification. So in our experiments, PCA is used for reducing the dimension of the image, then the reduced dimensions are treated as features for training classifier. For reducing dimension **Auto-Encoder** (deep learning based model) can also be used, but Auto-Encoder also requires separate training. **A part** from dimensionality reduction, there is also a scope of extracting **features (like corner, edges and SIFT points)** from the images and then use these extracted features for training classifier. Also fully connected Neural-Network (NN) or CNN (deep learning based models) can also be used for extracting features from image.

Deep nets is also doing well in the field of object detection. Rather than working on feature extraction and then training using SVM, Random forest or other methods. Deep learning models like NN, CNN or Faster-RCNN can be used

directly without any feature extraction. This will save time for training two separate models. In our application, there is a restriction on size of train dataset because these are manually marked on micrograph. So rather than training deep nets models from scratch, *transfer learning* can be used . **In transfer learning**, already trained model such as AlexNet, Vgg16 and ResNet are used as the base network. Before training, some layers are tweaked or changed or removed depending on the application and train dataset size. So in our case, transfer learning concept can be used for training deep nets models.

There are several areas that can be explored for increasing the efficiency (time and accuracy) of detector. Also, testing detector on actual dataset rather than simulated dataset because in the end detector has to be used in real life application.

Bibliography

- [1] Jacob Bruggemann, Gabriel C Lander, and Andrew I Su. Exploring applications of crowdsourcing to cryo-em. *bioRxiv*, page 220145, 2017.
- [2] Yuxiang Chen, Thomas Hrabe, Stefan Pfeffer, Olivier Pauly, Diana Mateus, Nassir Navab, and Friedrich Förster. Detection and identification of macromolecular complexes in cryo-electron tomograms using support vector machines. In *Biomedical Imaging (ISBI), 2012 9th IEEE International Symposium on*, pages 1373–1376. IEEE, 2012.
- [3] CHRISTOPH DIEBOLDER, Abraham J Koster, Roman I Koning, et al. Pushing the resolution limits in cryo electron tomography of biological structures. *Journal of microscopy*, 248(1):1–5, 2012.
- [4] Eeshan Malhotra and Ajit Rajwade. Tomographic reconstruction from projections with unknown view angles exploiting moment-based relationships. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1759–1763. IEEE, 2016.
- [5] Frank Natterer. *The mathematics of computerized tomography*. Siam, 2001.
- [6] Sjors HW Scheres. Relion: implementation of a bayesian approach to cryo-em structure determination. *Journal of structural biology*, 180(3):519–530, 2012.
- [7] Fred J Sigworth. Principles of cryo-em single-particle image processing. *Microscopy*, 65(1):57–67, 2016.
- [8] Hong-Wei Wang and Jia-Wei Wang. How cryo-electron microscopy and x-ray crystallography complement each other. *Protein Science*, 26(1):32–39, 2017.